IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 3, 2022, accepted June 13, 2022, date of publication June 17, 2022, date of current version June 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3184031

Video Sparse Transformer With Attention-Guided Memory for Video Object Detection

MASATO FUJITAKE[1], **(Graduate Student Member, IEEE),**
AND AKIHIRO SUGIMOTO[2], **(Member, IEEE)**
[1]Department of Informatics, The Graduate University for Advanced Studies, SOKENDAI, Miura-gun 240-0193, Japan
[2]National Institute of Informatics, Tokyo 101-8430, Japan

Corresponding author: Masato Fujitake (fujitakemasato@gmail.com)

ABSTRACT Detecting objects in a video, known as Video Object Detection (VOD), is challenging since appearance changes of objects over time may bring detection errors. Recent research has focused on aggregating features from adjacent frames to compensate for the deteriorated appearances of a frame. Moreover, using distant frames is also proposed to deal with deteriorated appearances over several frames. Since an object's position may change significantly at a distant frame, they only use features of object candidate regions, which do not depend on their position. However, such methods rely on object candidate regions' detection performance and are not practical for deteriorated appearances. In this paper, we enhance features element-wisely before the object candidate region detection, proposing Video Sparse Transformer with Attention-guided Memory (VSTAM). Furthermore, we propose aggregating element-wise features sparsely to reduce processing time and memory cost. In addition, we introduce an external memory update strategy based on the utilization of the aggregation to hold long-term information effectively. Our method achieved 8.3% and 11.1% accuracy gain from the baseline on ImageNet VID and UA-DETRAC datasets. Our method demonstrates superior performance against state-of-the-art results on widely used VOD datasets.

INDEX TERMS Video object detection, video analysis, object detection.

I. INTRODUCTION

Video object detection (VOD) extends still image object detection [1], [2] into videos. Applying still image object detectors suffers from stably detecting objects in a video due to appearance changes of objects over time. A video has rich temporal information, in which the same object may appear in multiple frames for a certain period. Therefore, incorporating temporal information into the detectors has thus been proposed to improve accuracy. The mainstream approach in recent years is feature refinement, considering the spatiotemporal information [3]–[5]. It aggregates useful features from surrounding frames to compensate for the deteriorated features of the target frame. FGFA [3] and MANet [6] proposed to utilize optical flow whereas TSSD-OTA [7] and some methods [8], [9] exploited recurrent neural networks to propagate features from neighboring frames.

Recently, leveraging distant frames from the target frame has been proposed [4], [5], [10] because considering only the neighboring frames suffers from detection on deteriorated apparent frames that persist for a while [3], [6], [7]. To utilize distant frames, object misalignment becomes an issue due to significant object position changes. Thus, focused on is the aggregation of object candidate region features generated from Region Proposal Networks (RPN) [1]. It allows us to aggregate features independent of the object positions; however, it cannot suppress false-negative detection since object

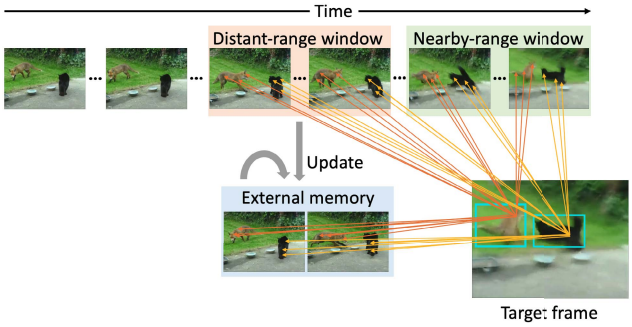


FIGURE 1. An example of the behavior of VSTAM framework. It aggregates related information from past frames spatiotemporally to refine the target frame's features, including ones in external memory. The orange and yellow arrows represent the highly related positions between frames.

The associate editor coordinating the review of this manuscript and approving it for publication was Yun Zhang.

candidates are assumed to be detected. In order to adapt detection miss, feature refinement before object candidate detection is necessary. Besides, the memory consumption cost becomes crucial to leverage distant frames if a static sliding window or external memory is used where it is updated randomly [11] or in order [5], [12]. Accordingly, adaptively holding the most vital frame features based on utilization of aggregation is preferable.

Motivated by the above observations, we propose Video Sparse Transformer with Attention-guided Memory (VSTAM) that refines features at the element level sparsely, considering both nearby and distant dependencies. It refines features element-wisely by considering all features in sampled frames before the object candidate detection. To avoid time-consuming and memory-intensive refinement, we propose a sparse aggregation method, taking into account the redundancy of a video. Moreover, VSTAM possesses an external memory that adaptively holds the most vital frame features. Figure 1 illustrates the concept of the proposed method. The proposed method aggregates features associated with each element widely and appropriately from multiple locations and frames with sparse attention, and retains the more vital frame features sequentially in the external memory. Even if features of some objects or frames are degraded, it appropriately uses features from other locations in other frames for aggregation. Moreover, if valuable frame features are in the sliding window, they can be updated into the external memory and aggregated. Despite its simplicity in structure, VSTAM outperforms existing methods surprisingly on ImageNet VID [13] and UA-DETRAC [14].

The contribution of our paper is listed as follows.

- We propose a spatiotemporal feature enhancement framework with adaptive updated external memory for video object detection to sparsely refine the features at the element level, considering both nearby and distant dependencies.
- To realize element-level aggregation efficiently, we propose a video sparse transformer to learn the aggregation of sparsely distributed features in space and time.
- With a simple but effective feature enhancement framework, we achieved SoTA using ResNet-101 on online settings with 85.7 mAPs on ImageNetVID and 90.39 AP on UA-DETRAC. We also achieved superior accuracy improvement than SoTA without difficulty on the more complex Youtube-VIS dataset.

II. RELATED WORK

A. VIDEO OBJECT DETECTION

VOD is an extended task of still-image object detection [1], [15], [16] to tackle video issues such as appearance changes over time. It can be categorized into two groups: box- and feature-level methods.

Box-level methods exploit tracking [17] and tubelet [18], [19] to associate related boxes and IoU [20] over time to create temporal links. Despite improvements, they need to detect objects in most frames to associate detection results. Besides, they cannot be trained in an end-to-end manner, or they require high computational costs.

Feature-level methods, on the other hand, enhance detection frame features with surrounding ones. According to the temporal duration to utilize, they can be divided into two subcategories: short- and long-term feature refinement.

Short-term feature refinement methods exploit optical flow [3], [6], [21], recurrent neural network [7], [8], and attention mechanism [22]–[25]. Although they can improve the detection by using nearby frames to enhance the whole features, the significant misalignment between frames impedes feature refinement when distant frames come in. Therefore, it is difficult to deal with video issues such as motion blurring lasting for multiple frames.

Long-term feature refinement methods [4], [5], [10] leverage distant frames to overcome multiple deteriorated frames. They mainly consider object-level features robust against object misalignment between frames. SELSA [4] considers the semantic impact between related object candidate regions in all the frames. RDN [10] distills relation through repeatedly refining supportive object proposals with high confidence. EBFA [26] proposes a temporal and spatial alignment module to refine object-wise features. Furthermore, HVR-Net [27] proposes to consider the relation of object-level features among different videos. MEGA [5] considers both local and global aggregation in time to enhance the feature representation. These methods, however, employ object-wise aggregation after the region proposals, which heavily rely on detection accuracy. On the contrary, our proposed method refines the whole features element-wisely, considering spatiotemporal information before detection. Table 1 briefly summarizes these methods.

B. EXTERNAL MEMORY FOR VOD

External memory has been studied in the feature-level method and can be classified into two categories based on updating strategy. The first category employs the first-in, first-out strategy, which utilizes the memory to extend past features [5], [12]. However, they do not consider the importance of each feature to keep.

The second one dynamically updates the external memory depending on the specific sampling strategy, and our approach falls into this category. OGEMN [11] proposes a top-down object-guided strategy, which computes features that give a high confidence level belonging to the detected objects and selects the higher ones to store. MAMBA [28] employs a random sampling strategy considering video redundancy and proposes feature-wise deleting to remove redundant features for efficient computation. Our method differs from the above methods in that it selects features based on the sum of the attention weights of the aggregated elements at the frame level in a more straightforwardly bottom-up manner. We summarize these methods in Table 2.

TABLE 1. Highlights and limitations of various long-term feature refinement methods.

Methods	Methodology	Highlights	Limitations
RDN [10]	Refining features after RPN based on the object's relation.	Feature refinement, regardless of object misalignment across frames.	Fixed sliding window.
SELSA [4]	Refining features after RPN considering a whole video sequence.	Tolerance to appearance changes of objects for a while.	Ignoring the balance of nearby and distant frames.
EBFA [26]	Refining features after RPN temporal and spatial separately.	More precise feature alignment among frames.	Fixed sliding window.
MEGA [5]	Refining features after RPN by separating nearby and distant frames.	Considering the balance of nearby and distant frames.	Difficult to deal with false-negative detections.
HVR-Net [27]	Refining features considering the relation of multiple videos.	Mutual correction with multiple videos.	Difficult to deal with false-negative detections.

C. TRANSFORMER

Transformer [29] is an architecture for learning sequential data dependency. The vanilla transformer [29] and its similar one, non-local [30], are powerful models; however, they suffer from computational costs when they come to large tensors due to the considerable sequence length and resolution. Some attempts are reported to reduce the cost by making the transformer's self-attention map sparse [31]–[33]. Sparse masks, such as slide windows, enable a transformer to abbreviate computation on no mask [31], [32]. Although these masks work well on NLP tasks [31], [33] and a single image [32], we cannot directly apply them to the video sequence due to spatial and temporal constraints. Our proposed *video sparse attention* properly captures long-range dependencies in space and time with reasonable computational cost and memory consumption.

Recently, TimeSformer [34] and SSTVOS [35] have been proposed to make a transformer sparse for video classification and video object segmentation, respectively. However, TimeSformer considers only the element-wise temporal information at the same position over multiple frames. On the other hand, SSTVOS considers only temporally and spatially local elements. Hence, they are vulnerable to significant object motion. On the contrary, our method also considers the object moving in distant frames by incorporating randomness. We summarize the highlights and limitations of these methods in Table 3.

DETR [16] and Deformable DETR [36] have recently been proposed for still-image object detection by utilizing Transformer. Our method differs from these methods because it considers spatiotemporal information for feature enhancement. Furthermore, their extension to VOD has been proposed [37]. However, since it neither considers temporal information for feature refinement nor leverages distant frames adaptively, the temporal information is not effectively utilized.

III. PROPOSED METHOD

Our proposed VSTAM is depicted in Figure 2. It consists of five components: feature embedding with frame selection, encoder, decoder, detection modules, and external memory. First, to feed short- and long-range temporal information, we effectively collect both nearby and distant frames. The feature embedding module then extracts features. The features are further compressed and flattened into one dimension [16]. Then, they are concatenated in the timeline order with features in the external memory to have a one-dimensional sequence. Then, the sequence together with positional encoding is passed to the encoder module to exploit the long-range sequential dependency among frames. Next, the high-level encoded features and the positioned frame query are passed into the decoder module for aggregation to obtain enriched features. They are passed to the detection module for object detection. Finally, the features to be kept are selected for the next-frame detection. The external memory is updated based on the attention weight of each frame in the decoder to utilize the features of essential frames in the distant frame window.

A. FRAME SELECTION FROM SHORT- AND LONG-RANGES

Given video frames $\{I_t\}_{t=1}^{T} \in \mathbb{R}^{H_0 \times W_0 \times C_0}$, where T is the length of a video and H_0, W_0 and C_0 respectively denote height, width, and number of channels, our goal is to detect objects in the current frame (at time k) I_k with reference frames R_k where $|R_k| = m$ for a given m. Reference frames are used for aggregation to have the enriched feature of the current frame.

To capture the long-term temporal dependencies of a video, we need to collect reference frames from short- and long-term periods. For a given m (the number of past frames used for aggregation), we define the set S_{sparse} of differences of time from the current frame time as follows: $S_{sparse} = \{2^i | 0 \leq i < m\}$ (Fig. 3b). We then define $R_k = \{I_{k-n} | n \in S_{sparse}\}$. In this way, we can effectively collect nearby and distant frames as reference frames. I_k and R_k are fed to the feature embedding module.

Our collected reference frames consist of nearby frames that complement the blur in a short time temporal densely and distant frames that are less affected by rare poses temporal sparsely. Compared with the standard dense sampling [3] (Fig. 3a), our collection way allows us to obtain a broader range of information with the same number of reference

TABLE 2. Highlights and limitations of various VOD methods with adaptive external memory.

Methods	Methodology	Highlights	Limitations
OGEMN [11]	Updating memory based on object features contributes to detection.	Considering the importance of each feature.	Complex and many calculations involved.
MAMBA [28]	Updating memory based on randomness.	Fast and simple strategy considering video redundancy.	Ignoring the importance of each feature.

TABLE 3. Highlights and limitations of various sparse transformer for video task.

Methods	Methodology	Highlights	Limitations
TimeSformer [34]	Aggregating features in the current frame and ones in element-wise temporal direction at the same position.	Consideration of long-term temporal information.	Vulnerable to object misalignment across frames.
SSTVOS [35]	Aggregating adjacent spatiotemporal features around each element.	Simple sparse aggregation.	Ignoring global spatiotemporal information in a video.

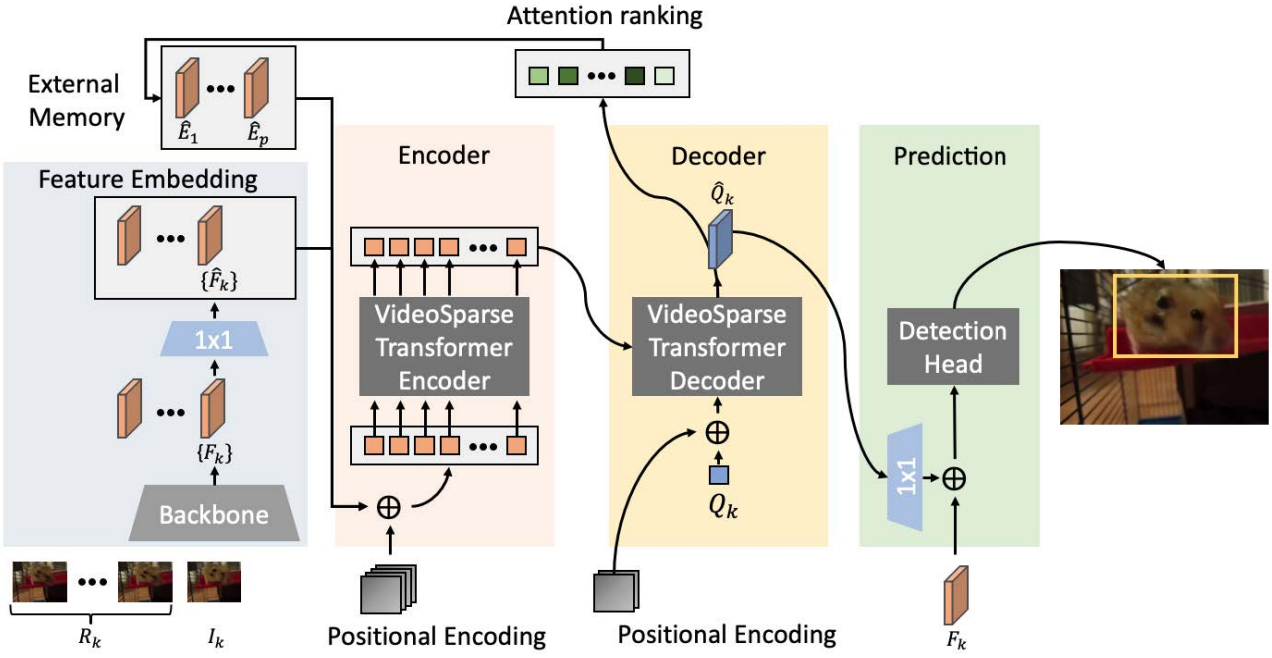


FIGURE 2. The architecture of the proposed Video sparse transformer with attention-guided memory (VSTAM). It receives the detection frame and the sparsely sampled reference frames in time. It then compresses the features from the backbone, and converts them into a 1D sequence along with the ones in external memory. The encoder spatiotemporally samples the elements sparsely, and the decoder outputs the enhanced features for detection. In order to update the external memory based on the importance of features, the weighting of each element at the aggregation is accumulated for each frame. Finally, the frame's features with the highest weight are kept in the external memory in order.

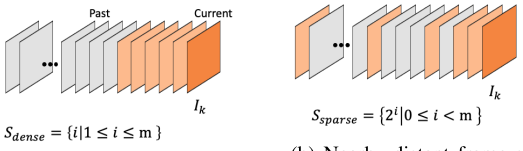
frames. We remark that we use "nearby frames" to refer to the last five frames and "distant frames" to refer to the frames after that since in existing works [3], [12] focusing only on nearby frames, the weights are applied only up to five frames.

B. FEATURE EMBEDDING

Given the selected frames R_k and I_k, the feature embedding module extracts features $\{F_k\}$. We utilize a shared-weighted ResNet [38] or ResNeXt [39]. Following DETR [16], we use a 1×1 convolution to reduce the channel dimension of features $\{F_k\} \in \mathbb{R}^{H \times W \times C}$ from C to a smaller dimension d, creating new features $\{\hat{F}_k\} \in \mathbb{R}^{H \times W \times d}$. We then collapse the spatial dimensions of $\{\hat{F}_k\}$ into one dimension, resulting in $HW \times d$ features. The external memory contains additional flattened features $\{\hat{E}_q\}_{q=1}^{p}$. The newly sampled features and the ones stored in the external memory are concatenated in the order of the timeline. The number of frames is $L = p+m+1$. In this way, we obtain a feature sequence $Z \in \mathbb{R}^{LHW \times d}$.

C. VIDEO SPARSE TRANSFORMER

Based on the vanilla transformer [29], we develop video sparse transformer (VST) so that it aggregates information from multiple frame features. The vanilla transformer exploits a self-attention mechanism to learn the elements' dependencies and gather information for an input sequence. Although a vanilla transformer considers all the elements,



(a) Nearby-only frame selection (b) Nearby-distant frame selection

FIGURE 3. The reference frame selection for feature aggregation from a video clip. *m* is the number of past frames used for aggregate. Dark and light orange indicate the current frame and the selected reference ones, respectively.

considering all the elements of a video sequence is unnecessary because of the redundancy involved in the video (ex., objects may appear at similar positions for a certain period in multiple frames). We thus follow recent work [32], [33] that makes self-attention sparse and samples elements more efficiently for VST.

1) VIDEO SPARSE ATTENTION

To realize video sparse attention, the video sparse attention masking operation $M(\cdot)$ is implemented on self-attention [29] with the below modification. The modified formulation of a uni-head sparse self-attention is

$$\text{SparseAttention}(Q, K, V) = \text{softmax}(M(\frac{QK^\top}{\sqrt{d_k}}))V, \quad (1)$$

where $K \in \mathbb{R}^{l \times d_k}$, $V \in \mathbb{R}^{l \times d_v}$, $Q \in \mathbb{R}^{l \times d_k}$ are the key, value, and the query, respectively. l is the length of input sequence, d_v and d_k are the embedding dimensions of the value and key. A sparse mask $M \in [0, 1]^{l \times l}$ is defined as

$$M(i, j) = \begin{cases} 1 & \text{if query } i \text{ attends to key } j, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Video sparse attention is designed taking into account the following considerations. First, to refer to all the elements in a frame spatial locally and globally, we introduce the frame attention (Fig. 4a). It allows the self-attention to refer only to each frame's elements, thus improving the features considering its spatial context. To compensate lack of temporal information, we introduce two sparse masks: random and position attention.

Random attention (Fig. 4b) masks a certain percentage of the elements, allowing access to a wide range of features. Different from the original one [31] in NLP, we mask each frame with a random probability r instead of the entire sequence because a video is divided into frames.

Although random attention enables us to obtain information from multiple frames, it cannot sometimes aggregate features reliably when objects remain in a specific area over multiple frames. To reliably extract information from around the same location over multiple frames, we introduce position attention (Fig. 4c). It plays the role of aggregating features from the corresponding location in the temporal direction. It only considers the same position at each frame; it is sensitive to object motion. Therefore, we applied a mask like a 3×3 dilated convolution kernel (Rate $= 2$) [40] to each element to give the position attention to a wide field of view for robustness against object motion.

The combined masks of the frame, random, and position are video sparse attention (Fig. 4d) and applied to the self-attention of transformer [29]. We exploit this sparse transformer for both the encoder and decoder.

2) ENCODER AND DECODER

The encoder and decoder follow the original layered architecture of the transformer [29] except for its self-attention. We replace the standard self-attention with video sparse attention. Given the positioned and flattened feature sequence Z, we obtain the embedded sequence $\hat{Z} \in \mathbb{R}^{LHW \times d}$ via the encoder.

The function of the decoder is to generate the refined features. To decode at each element of the features, a query sequence $Q_k \in \mathbb{R}^{HW \times d}$ is required and obtained by flattening embedded features $\hat{F}_k$. Then, the decoder outputs the enriched feature sequence $\hat{Q}_k$ using the query $\hat{Z}$ and embedded Q_k sequence.

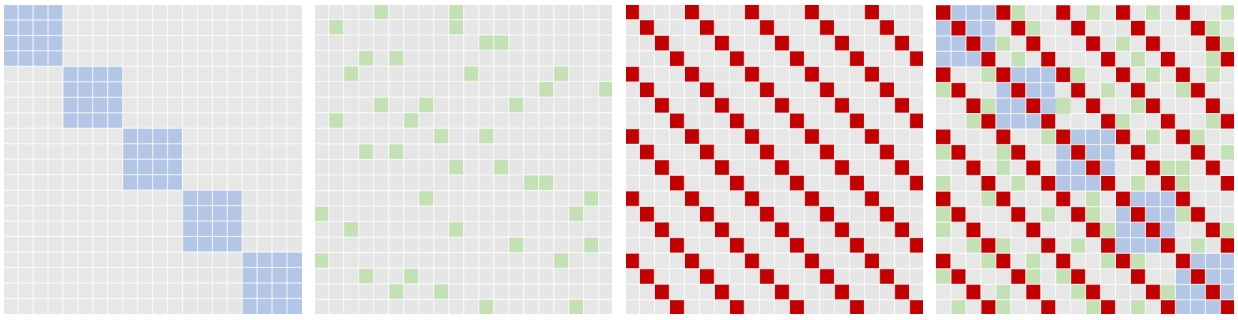
D. DETECTION

Thanks to the decoder, we have refined sequence $\hat{Q}_k$. To utilize it for detection, we expand its spatial dimension. It contains the video sequence's spatiotemporal local and global information; however, it loses the detailed information due to the compression of 1×1 convolution operation in the feature embedding process. Therefore, we decompress $\hat{Q}_k$ with 1×1 deconvolution operation in the channel direction to generate the features $\tilde{Q}_k \in \mathbb{R}^{H \times W \times C}$. Then, we merge both features $\tilde{Q}_k$ and F_k by the element-wise sum to acquire the final refined features for detection.

E. EXTERNAL MEMORY

To adaptively store the features of the most vital frames in the external memory, we select them based on the importance of each frame. We regard it according to the attention weights, which are already computed when VST aggregates each element. The element-level attention weights are accumulated for each frame to measure the importance of each frame. This is the "attention ranking" shown in Fig. 2, and we keep up to the p-th features as $\{\hat{E}_q\}$ in the external memory, arranged in the cumulative order of attention weights. We have two types of feature candidates stored in the external memory. One is the features kept in the current frame's external memory. The other is the features of the distant frames newly loaded in the sliding window. The former features are from distant frames deemed vital and stored at the time of the previous detection. The latter features are from newly sampled distant frames and stored at the time of the current detection. This design enables us to handle the issue that the critical features in the past are not always valid due to video scene changes.

The distant frames are defined in section III-A. The role of the external memory is to hold long-term features and deal with scenes that are difficult to detect by using only neighboring frames, so adjacent frames are not to be stored.



(a) Frame (b) Random (c) Position (d) Video sparse

FIGURE 4. Visualization examples of the sparse attention. Here, we assume a video with five consecutive frames, each frame possessing a 2×2 feature element. Gray color indicates the absence of attention. (a) frame-wise attention, which cares only in self-frame, (b) random attention, (c) Position attention, which focuses on the same position of each frame, (d) the combined attention map of Video Sparse Transformer.

IV. EXPERIMENTS

A. DATASETS AND METRICS

We utilized the two datasets shown in Table 4 for our validation.

1) IMAGENET VID

ImageNet VID [13] is a large-scale benchmark for video object detection. It is one of the challenging datasets to detect objects because it offers a wide variety of appearance changes of objects over time. It has 30 categories and contains 3,862 training and 555 validation videos with 25 and 30 frame rates. We evaluate our method on the validation set and use the mean average precision (mAP) following widely adopted protocols in [3].

2) UA-DETRAC

UA-DETRAC [14] is a large-scale benchmark for real-world traffic scenes. It contains 60 videos in the training set and 40 videos in the test set. The videos are recorded at 25 fps, with a resolution of 960 × 540 pixels. We validate our method on the test set and use the average precision (AP) at IoU threshold 0.7 as the evaluation metric for precise localization.

B. NETWORK ARCHITECTURE

1) FEATURE EXTRACTOR

We utilize ImageNet pretrained ResNet-50 [38], ResNet-101 for detail analysis and performance comparison, respectively. Following a common practice [3], [6], we enlarge the resolution of features by modifying the stride of the first convolution block in the last stage of convolution, *conv5*, from 2 to 1. We also set the dilation of these convolutional layers to 2 to retain the receptive field size.

2) DETECTION NETWORK

We exploit Faster R-CNN [1] as the detection module. For a fair comparison, we follow the commonly employed setting [4], [5]. Specifically, we leverage 12 anchors with 4 scales $\{64^2, 128^2, 256^2, 512^2\}$ and 3 aspect ratios $\{1:2, 1:1, 2:1\}$ for regression and classification. During training and inference, 3,000 and 300 candidate boxes are generated in previous and post non-maximum suppression (NMS), respectively.

3) VSTAM

We set the frame selection (Fig. 3b) at training and inference stages with temporal window size of $m = 5$. Unless otherwise stated, we conducted our experiments with external memory set to $p = 2$. Therefore, we utilize $L = 8$ frames. In the embedding process, we set the compressed dimension d of the features to 128. In VST, we utilize multi-head attention with the number of heads $h = 8$. The number of layers in the encoder and decoder is set to 4, respectively. We use a sinusoidal positional encoding. For the sparse attention, we set a random ratio with $r = 10\%$ at each frame.

C. IMPLEMENTATION DETAILS

We implemented VSTAM on detectron2 [41] and performed on two Titan RTX GPUs with AdamW [42]. For ImageNet VID, we train VSTAM on a combination of ImageNet VID and DET [13] following common protocols in [4], [5], [43]. In DET, we select the same 30 classes as in the VID dataset and follow the data augmentation strategy proposed in [4]. For UA-DETRAC, we utilize only its training dataset.

The input images are resized to have their smaller side to be 600 pixels. If there is no specified frame at the frame selection, zero padding is performed. In addition to the frames for the fixed window, p frames are randomly sampled from the video clip for the external memory. Each GPU holds two mini-batches, and each mini-batch contains one set of images or frames. The model with a vanilla transformer was trained with one mini-batch due to memory constraints and adjusted the hyperparameters according to the batch size [44]. We employ a base learning schedule as $1\times$ for 13 epochs with learning rate decay, dividing by ten at epochs 9 and 12, respectively. We utilize the $1\times$ learning schedule for component analysis and the $3\times$ one to compare the competitive models, based on the observation [45]. The initial learning

TABLE 4. Overview of datasets.

Name	Number of videos		Number of classes	Evaluation metrics
	Train set	Evaluation set		
ImageNet VID [13]	3,862	555	30	mAP (IoU=0.5)
UA-DETRAC [14]	60	40	1	AP (IoU=0.7)

rate is set to 10^{-4}. At inference, an NMS of 0.5 IoU threshold is adopted to suppress reduplicate detection boxes.

D. COMPARISON WITH THE STATE-OF-THE ARTS

1) COMPARISON ON IMAGENET VID

We employ ResNet-101 [38] and ResNext-101 [39] as a feature extractor for a fair comparison. We compare the models separately since the accuracy differs among offline, online, and post-processing cases. For comparison, we show the offline setting results for the case where five frames extend the sliding window into the future ($L = 13$).

Table 5 shows the comparison result between state-of-the-art methods on online, offline and post-processing conditions. Among all the methods, VSTAM achieves the best performance on all backbone and conditions. With ResNet-101 backbone, our online model achieves 85.7% mAP, 1.1% absolute improvement over the recent and most powerful competitor MAMBA [28], which utilizes external memory. Compared with FGFA [3], MANet [6], and STSN [43], which aggregate element-wise feature from nearby frames, VSTAM outperforms more than 8 points. The gap between VSTAM and the above is feature aggregation with global spatiotemporal context. VSTAM also outperforms some methods [4], [5], [26], which utilize object-wise feature aggregation. The object-wise approaches provide effective improvement; however, they cannot refine features unless detect object candidate regions are given. Our method considers element-wise features from distant and nearby frames before the region proposal network, leading to the best performance on ImageNet VID. Figure 5 shows the detection results of Faster R-CNN, FGFA, MEGA and VSTAM. We see that VSTAM improves the detection of even severely damaged scenes.

By replacing the backbone from ResNet-101 to ResNeXt-101, our model achieves a better performance of 87.0% mAP, as expected. In an offline setting, our model achieves an accuracy of 87.6%. Moreover, we applied post-processing to the offline model as many offline methods do. For the post-processing method, we adopt Seq-NMS [20], which refines scores of weaker detection from nearby frames. Our method still performs the best, obtaining 86.4% and 88.1% mAP with backbone ResNet-101 and ResNeXt-101, respectively.

Furthermore, TransVOD++ [37], which extends Deformable DETR [36] spatio-temporally, employs the strong backbone [51] and detector [36]. To fairly compare with it, we replace our backbone and detector with them; our model reaches 91.1% mAP in the offline setting, 1.1 points higher than TransVOD++. We see that element-wise feature refinement and attention-guided external memory are essential to detect objects stably.

Table 6 shows accuracy and speed comparison on the same architecture and GPUs. We can see that VSTAM is superior in accuracy while the speed is faster than most methods. Next, we replaced the detector with R-FCN [50] to compare the performance of the methods with the external memory under the same conditions and GPU. As shown in Table 7, we confirm that VSTAM is superior in both accuracy and speed. OGEMN and MAMBA have two-step frame-wise and object-wise aggregation based on complex update and delete rules, requiring more processing time. On the other hand, VSTAM deals only with features element-wisely and reduces run-time by a simple rule that holds the features most used in the enhancement.

2) COMPARISON ON UA-DETRAC

The detection results on the UA-DETRAC dataset are reported in Table 8. YOLOv3-SPP, MSVD_SPP, and SpotNet are still image detectors, and they propose to improve accuracy by detector modification and introducing spatial attention mechanisms. In contrast, TFEN and FFAVOD-SpotNet utilize temporal information to refine features. We remark that the methods on UA-DETRAC cannot be fairly compared because they use different feature extractors and detectors.

Our online model with ResNet-101 achieves 90.39% AP, 2.29% absolute improvement over FFAVOD-SpotNet [55], which employs a strong base detector [2] and a feature refinement module to fuse multiple nearby frame features channel-wisely in offline settings.

E. ERROR ANALYSIS

Using TIDE [56], we analyze what types of errors in VOD are resolved. TIDE classifies object detection errors into misclassification, incorrect localization, duplicate detection, false-positive detection, and miss. We performed error analysis on the ImageNet VID dataset to use the officially published model [5]. Figure 6 shows the error results of the Faster R-CNN [1], MEGA [5] and VSTAM, where the horizontal axis shows the error items and the vertical axis shows the amount of error accumulation proposed in TIDE [56]. We see that Faster R-CNN produced many "Cls", "Bkg", and "Miss" detection errors. This is due to the appearance changes of objects over time in videos. In contrast, VSTAM significantly reduces the amount of errors, especially for "Bkg" and "Miss", compared to MEGA. In addition, our element-wise aggregation also reduces the class error. We see that it is crucial to enhance the features before object candidate detection for VOD.

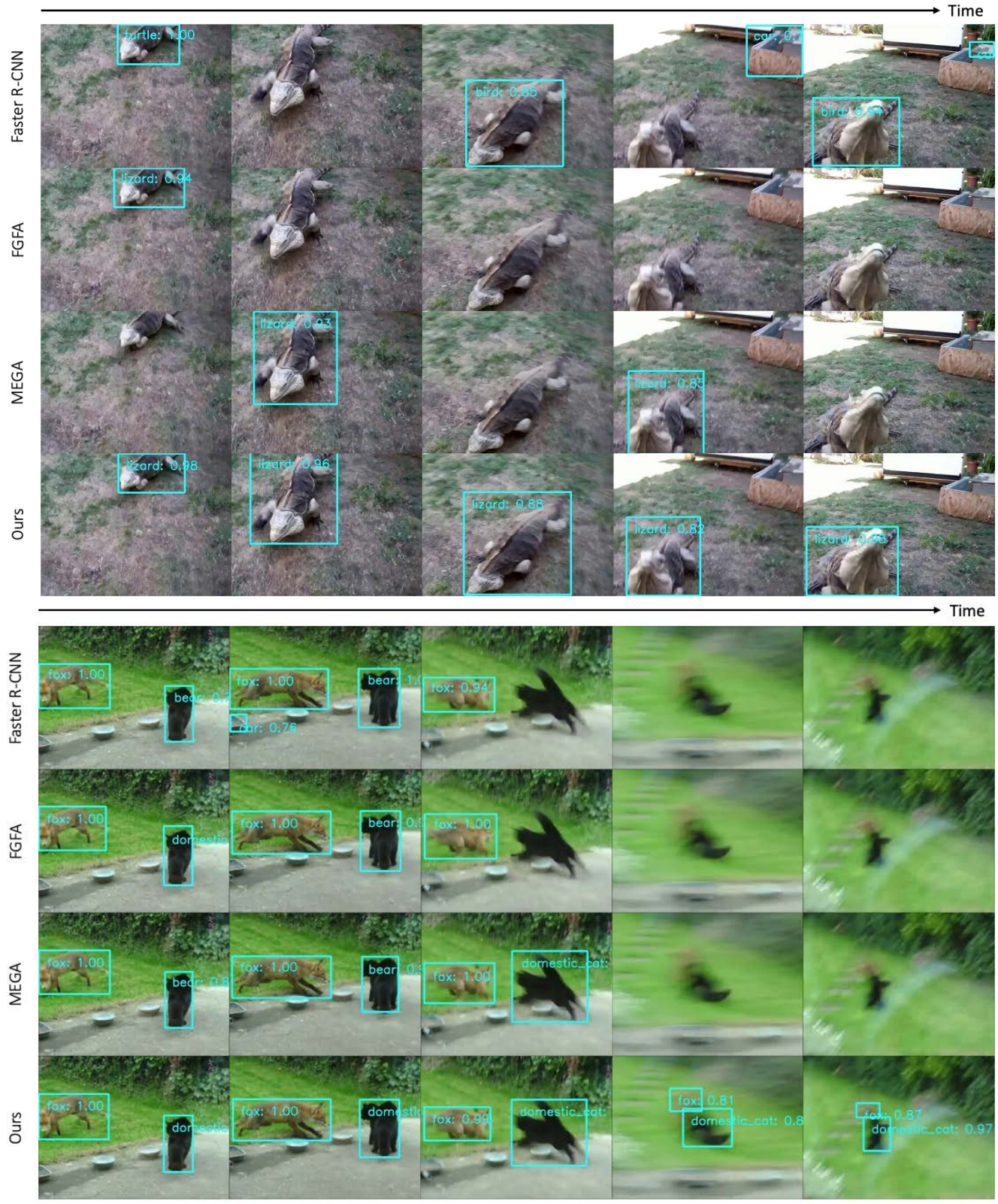


FIGURE 5. Visualized comparison against the state-of-the-art methods on ImageNet VID. We show the detection result of our method against Faster R-CNN [1], FGFA [3] and MEGA [5]. FGFA refines features considering nearby frames, and MEGA incorporates distant frames with object candidate detection. All models use ResNet-101 [38] as the backbone and Faster R-CNN [1] as the detector. We show randomly sampled frames from a video clip. Our detection suppresses the false negative and false positive detection.

TABLE 5. Performance comparison on ImageNet VID.

Methods	Backbone	Base Detector	mAP		
			Online	Offline	Post-processing
OGEMN [11]	ResNet-101 + DCN	R-FCN	80.0	–	81.6
PLSA [23]	ResNet-101 + DCN	R-FCN	80.0	–	–
LSTS [24]	ResNet-101 + DCN	R-FCN	80.1	–	–
D&T [17]	ResNet-101	Faster R-CNN	80.2	–	–
LRTR [46]	ResNet-101	FPN	81.0	–	–
MAMBA [28]	ResNet-101	Faster R-CNN	84.6	–	–
VSTAM(Ours)	ResNet-101	Faster R-CNN	**85.7**	**86.2**(+0.5)	**86.4**(+0.2)
FGFA [3]	ResNet-101	R-FCN	–	76.3	78.4
MANet [6]	ResNet-101	R-FCN	–	78.1	80.3
STSN [43]	ResNet-101 + DCN	R-FCN	–	78.9	–
SELSA [4]	ResNet-101	Faster R-CNN	–	80.3	82.7
RDN [10]	ResNet-101	Faster R-CNN	–	81.8	83.8
TransVOD [47]	ResNet-101	Deformable DETR	–	81.9	–
TROI [48]	ResNet-101	Faster R-CNN	–	82.0	–
MEGA [5]	ResNet-101	Faster R-CNN	–	82.9	84.5
HVR-Net [27]	ResNet-101	Faster R-CNN	–	83.2	83.8
TF-Blender [49]	ResNet-101	Faster R-CNN	–	83.8	–
DSFNet [25]	ResNet-101	Faster R-CNN	–	84.1	–
EBFA [26]	ResNet-101	Faster R-CNN	–	84.8	–
LRTR [46]	ResNeXt-101	FPN	84.1	–	–
MAMBA [28]	ResNeXt-101	Faster R-CNN	85.4	–	–
VSTAM(Ours)	ResNeXt-101	Faster R-CNN	**87.0**	**87.6**(+0.6)	**88.1**(+0.5)
RDN [10]	ResNeXt-101	Faster R-CNN	–	83.2	–
MEGA [5]	ResNeXt-101	Faster R-CNN	–	84.1	85.4
HVR-Net [27]	ResNeXt-101	Faster R-CNN	–	84.8	85.5
DSFNet [25]	ResNeXt-101	Faster R-CNN	–	85.4	–
VSTAM(Ours)	Swin Transformer Base	Deformable DETR	–	**91.1**	–
TransVOD++ [37]	Swin Transformer Base	Deformable DETR	–	90.0	–

TABLE 6. Comparison of accuracy and runtime on ImageNet VID. All processing time is measured on Titan RTX with Faster R-CNN and ResNet-101. The speed is reproduced in [28].

Methods	SELSA [4]	RDN [10]	MEGA [5]	MAMBA [28]	VSTAM
mAP	80.3	81.8	82.9	84.6	**85.7**
Runtime (ms)	**91.3**	128.0	182.7	110.3	95.2

TABLE 7. Comparison of external memory methods using R-FCN [50] on ImageNet VID. All runtime is measured on RTX Titan. The speed is reported in [28].

Methods	Base [50]	OGEMN [11]	MAMBA [28]	VSTAM
mAP	73.8	79.3 (+5.5)	81.6 (+7.8)	**82.7** (+8.9)
Runtime (ms)	**46.7**	89.1	90.1	80.2

TABLE 8. Performance comparison on UA-DETRAC.

Method	Overall	Easy	Medium	Hard	Cloudy	Night	Rainy	Sunny
TFEN [12]	82.42	97.40	88.90	72.18	87.54	82.41	72.32	90.78
YOLOv3-SPP [52]	84.96	95.59	89.95	75.34	88.12	88.81	77.46	89.46
MSVD_SPP [53]	85.29	96.04	89.42	76.55	88.00	88.67	78.90	88.91
SpotNet [54]	86.80	97.58	92.57	76.58	89.38	89.53	80.93	91.42
FFAVOD-SpotNet [55]	88.10	97.82	92.84	79.14	91.25	89.55	82.85	91.72
VSTAM (ours)	**90.39**	**97.83**	**94.77**	**82.21**	**92.58**	**91.97**	**85.23**	**94.54**

F. COMPONENT ANALYSIS

To evaluate the effectiveness of each component in VSTAM, we conduct ablation studies with ResNet-50 and the 1× learning schedule (i.e., 13 epochs).

1) INVESTIGATION OF VSTAM

Table 9 lists the ablation result of our module variants. To confirm the effectiveness of VSTAM, we use Faster R-CNN [1] as the baseline. First, we can see that the introduction

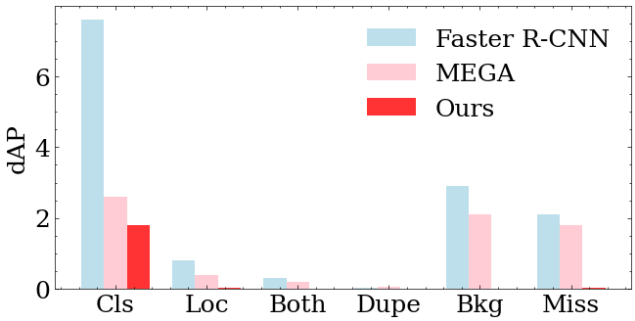


FIGURE 6. Visualized results of error analysis on ImageNet VID by TIDE [56]. Each bar indicates the amount of errors accumulated in each category. "Cls" represents that the model detected the object but misclassified it into another class. "Loc" means that the model detected the object with lousy localization. "Both" means occurring of both "Cls" and "Loc". "Dupe" represents duplicated detection for an object. "Bkg" means false positive, while "Miss" means that it does not detect the object even though an object exists there. Compared to MEGA, object-wise feature enhancement method, the proposed method significantly suppresses "BKG" and "Miss". Best viewed digitally and in color.

to VST brings significant gains to both datasets. We thus conclude that elemental aggregation is effective as feature aggregation for video. Next, we confirm that introducing the attention-guided external memory improves accuracy. Accordingly, all the factors are essential for VSTAM. Figure 7 shows the detection result comparison between the baseline [1] and VSTAM on UA-DETRAC. We confirm that VSTAM has improved detection results.

To check the effectiveness of sparse sampling in VSTAM, VST was replaced with a vanilla transformer. VST is about 1.2 points more accurate than the vanilla transformer on

TABLE 9. Impact of components in VSTAM.

Components		ImageNet VID	UA-DETRAC
Video Sparse Transformer	External Memory	mAP	AP
		71.7	73.1
✓		77.1	81.8
✓	✓	80.0	84.2
Vanilla transformer	✓	78.7	82.7

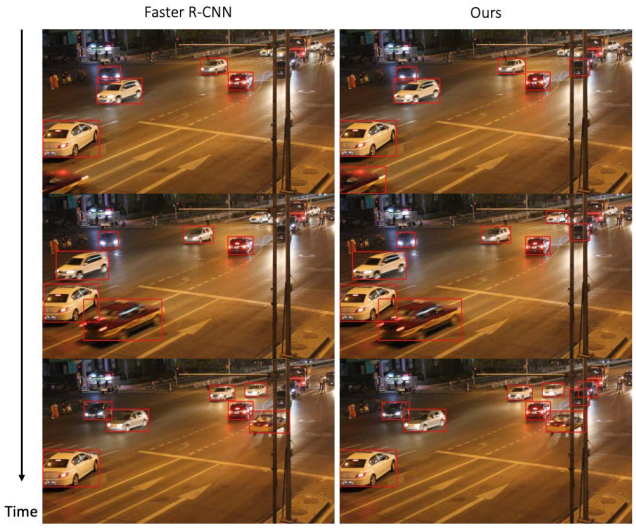


FIGURE 7. Visualized comparison between the baseline (Faster R-CNN [1]) and VSTAM (ours) on UA-DETRAC. We see that VSTAM detects targets robustly in the occluded scene.

TABLE 10. Impact of VSTAM to RPN on ImageNet VID.

VSTAM	AR_5	AR_{10}	AR_{100}
w/o	75.1	81.0	90.2
w/	79.2(+4.1)	86.1 (+5.1)	96.4 (+6.2)

both datasets. Indeed, we confirm that for video sequences, properly performing the sparse sampling achieves higher accuracy. Furthermore, VSTAM processes one frame in 52ms while the run-time of a vanilla transformer is 342ms. VSTAM (w/ VST) and VSTAM (w/ vanilla one) consume 2.1GiB and 7.2GiB memories per frame during inference. Additionally, 5.1GiB memories are required for each for Faster R-CNN. Our VST offers 658% speed up and 70.8% memory reduction thanks to our sparse sampling.

2) EFFECT OF FEATURE REFINEMENT TO RPN

The object-wise feature refinement methods, evaluated on ImageNet VID, employ features from RPN to deal with object misalignment among frames [4], [5], [10]. These methods rely on RPN detection, heavily degrading performance when RPN performance is not good. In contrast, our method refines the features before RPN. We evaluate how VSTAM affects the RPN in terms of Average Recall (AR). We select top $k = 5, 10, 100$ proposals generated by RPN and calculate AR_k. Table 10 shows the difference of Recall in RPN between the baseline model and the model with VSTAM. We can see that all the metrics are improved by the proposed method, confirming the effectiveness of the feature refinement before RPN.

TABLE 11. Performance comparison of feature aggregation modules.

Methods	Window-range		Imagenet VID	UA-DETRAC
	Nearby-only	Nearby-Distant	mAP	AP
VST (Ours)		✓	**77.1**	**81.8**
	✓		75.9	79.2
SSTVOS [35]		✓	74.4	75.0
	✓		74.7	75.6
Big Bird [31]		✓	72.4	73.7
	✓		72.1	74.1
TimeSformer [34]		✓	74.6	79.9
	✓		75.0	80.1

3) INVESTIGATION OF VIDEO SPARSE TRANSFORMER

We investigate VST from three aspects. They are sparse aggregation, video sparse attention, and the random attention ratio.

a: EFFECT OF SPARSE FRAME SELECTION ON AGGREGATION MODULES

We examine the effect of element aggregation across different frame selections. VSTAM aggregates information from a wide range of frame spans. We investigate how the frame selection affects element-level aggregation. In this experiment, we exclude external memory. We also compare our results with the sparse attention-based aggregation method: SSTVOS [35], Big Bird [31] and TimeSformer [34]. Note that since there is no official implementation of SSTVOS, we reproduced it and obtained a result of 0.1 points higher than the reported score in the paper. We also note that the percentage of random attention in Big Bird is set to 10% for a fair comparison.

Table 11 shows the performance with the two types of frame selection, where "Nearby-only" and "Nearby-Distant" represents dense sampling (Fig. 3a) and sparse sampling (Fig. 3b), respectively. Although all methods improve accuracy over the baseline, SSTVOS and TimeSformer lose accuracy when exploiting far frames. Big Bird does not handle sequential information well when adapted to a video, resulting in lower scores. This will be because it is proposed for NLP tasks. VST, on the contrary, improves accuracy by utilizing distant frames rather than nearby ones.

Figure 8 shows the difference in detection results depending on the frame selection of our method. We see that the sampling method, including distant frames, is robust to apparent changes over time, such as motion blur.

b: INVESTIGATION OF SPARSE ATTENTION

Table 12 shows the accuracy impact with each sparse attention method between baseline and VST. Using only the frame attention leads to significant accuracy decrease. Accuracy using only the random attention or the position attention is insufficient. We see that combining the frame, random, and position attentions improve accuracy, meaning that each is necessary.

FIGURE 8. Visualized comparison of detection results by different reference frame selection on ImageNet VID. We utilized our proposed models in Table 11. We show the detection results for four consecutive frames in which motion blur occurs. The correct object labels on the first video sequence are "fox" and "domestic_cat". The second video sequence contains "squirrel" and "domestic_cat". We confirm that the sparse frame selection stabilizes detection.

TABLE 12. Performance comparison of sparse attention modules.

Sparse attention			Imagenet VID	UA-DETRAC
Frame	Position	Random	mAP	AP
✓			73.1	75.8
	✓		74.0	79.2
		✓	74.3	74.8
✓	✓		74.8	79.4
✓		✓	75.3	78.8
✓	✓	✓	**77.1**	**81.8**

TABLE 13. Impact of the ratio of random attention on ImageNet VID.

Random r (%)	0	5	10	15	30	50	70	100
mAP (%)	74.8	76.4	77.1	77.1	76.9	76.5	76.3	75.9

c: EFFECT OF RANDOM ATTENTION

We investigate the impact of the ratio using random attention. Table 13 shows the performances under different ratios (r%) using random attention. $r = 0\%$ is identical with using frame attention and position attention only in Table 12 while $r = 100\%$ is identical with using a vanilla transformer. We see that the accuracy is improved by increasing r from 5% to 10%, but it gradually decreases from 15% to 100%. This indicates that introducing random attention is effective for feature aggregation, but using random attention too much is not a good way.

TABLE 14. Impact of external memory.

Methods	Update Rule	Update candidate	Additional frames	Imagenet VID mAP	UA-DETRAC AP
External Memory	Attention	–	0	77.1	81.8
	Attention	Distant	1	78.8	83.6
	Attention	Distant	2	80.0	**84.2**
	Attention	Distant	3	**80.1**	84.0
	Attention	Nearby&Distant	2	78.4	82.8
	Random	Distant	2	77.9	82.0
Extended sliding window	–	–	1	77.5	82.2
	–	–	2	77.7	82.3

4) EFFECT OF ATTENTION-GUIDED EXTERNAL MEMORY

Table 14 summarizes the accuracy effects of changing the number of additional frames p. To evaluate the effect of adaptively selected features, we increased sliding window width m. Although the extended sliding window improves the accuracy, the gain of the external memory is more prominent. The number of additional frames of the external memory is saturated after about two frames. It is no longer possible to obtain a significant gain. Figure 9 shows the difference in detection results when using attention-guided external memory and sliding windows at the additional two frames. We confirmed that the adaptive external memory updates prevent class errors and false-negative detection.

By default, the update candidates of external memory are only distant frames. We confirm that when nearby frames are included in update candidates, the accuracy decreases compared with only distant frames. It is better to utilize

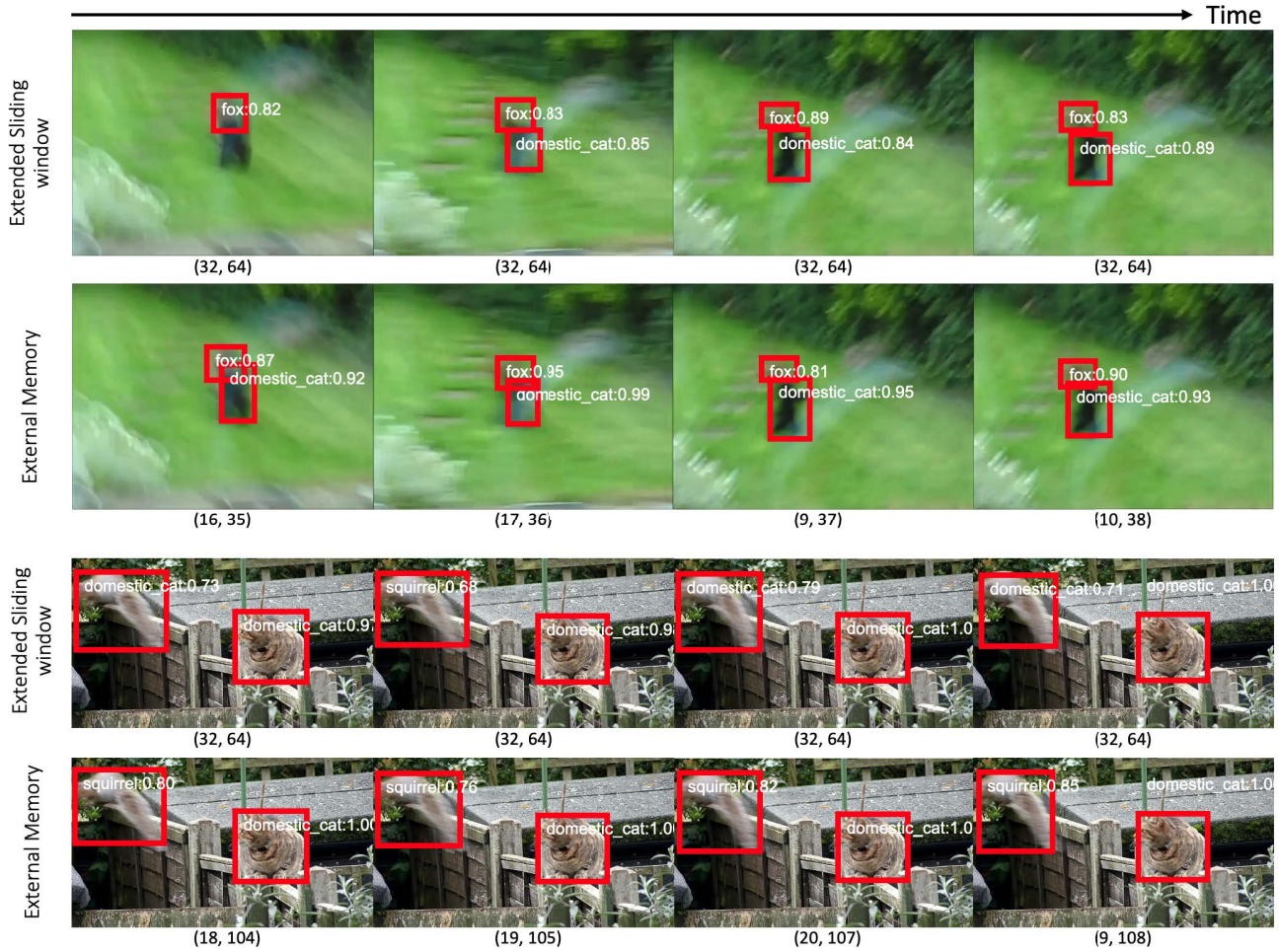


FIGURE 9. Visualized comparison of detection results between the attention-guided adaptive reference frame and static extended sliding window frame on ImageNet VID. We used the models in Table 14. We show the detection results for four successive frames in which motion blur occurs. The numbers below each sequence indicate the distance from the current frame to the additional reference frame. The proposed method reduces class errors by adaptively updating the external memory.

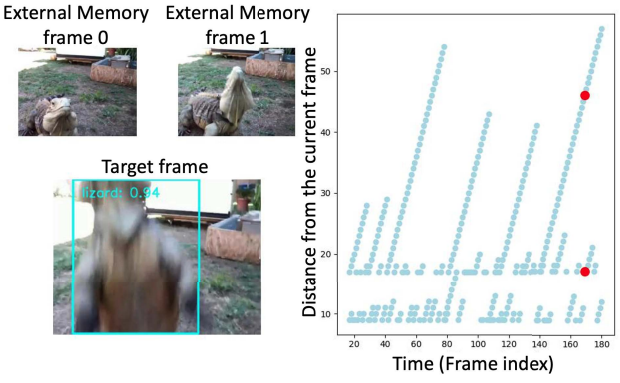


FIGURE 10. Visualized examples for the frames in the external memory and the current frames. The figure shows the distance to the frame stored in the external memory from the target frame. The horizontal axis shows the time of the video, the vertical axis shows the distance to the frames in the external memory, and the red dot shows the distance at that time. The left images show the target frame and the frame stored in the external memory at that time.

only distant frames as the candidates to utilize long-term information. We also show the case where the update frames are randomly selected to see whether the attention-guided update rule is valid. We see that the attention-guided updates are more accurate than random ones because they retain important frames adaptively.

Figure 10 shows how past features are stored in the external memory for a given video. It sometimes stores frames beyond 50 frames from the target frame. It is difficult to hold such a distant frame in a sliding window, confirming the importance of adaptive updates. Besides, the external memory tends to store the frames which capture objects clearly.

V. APPLICATION TO VIDEO INSTANCE SEGMENTATION

To validate the versatility of the proposed method, we applied our method to video instance segmentation (VIS), a combination of object detection, instance segmentation, and object tracking across frames. We evaluated our method on the YouTube-VIS dataset [57]. YouTube-VIS contains 40 object categories and consists of 2,238 training videos, 302 validation videos, and 343 test videos. The training is conducted on the training videos. Since the evaluation on the test set is currently closed, the evaluation is performed on the validation set.

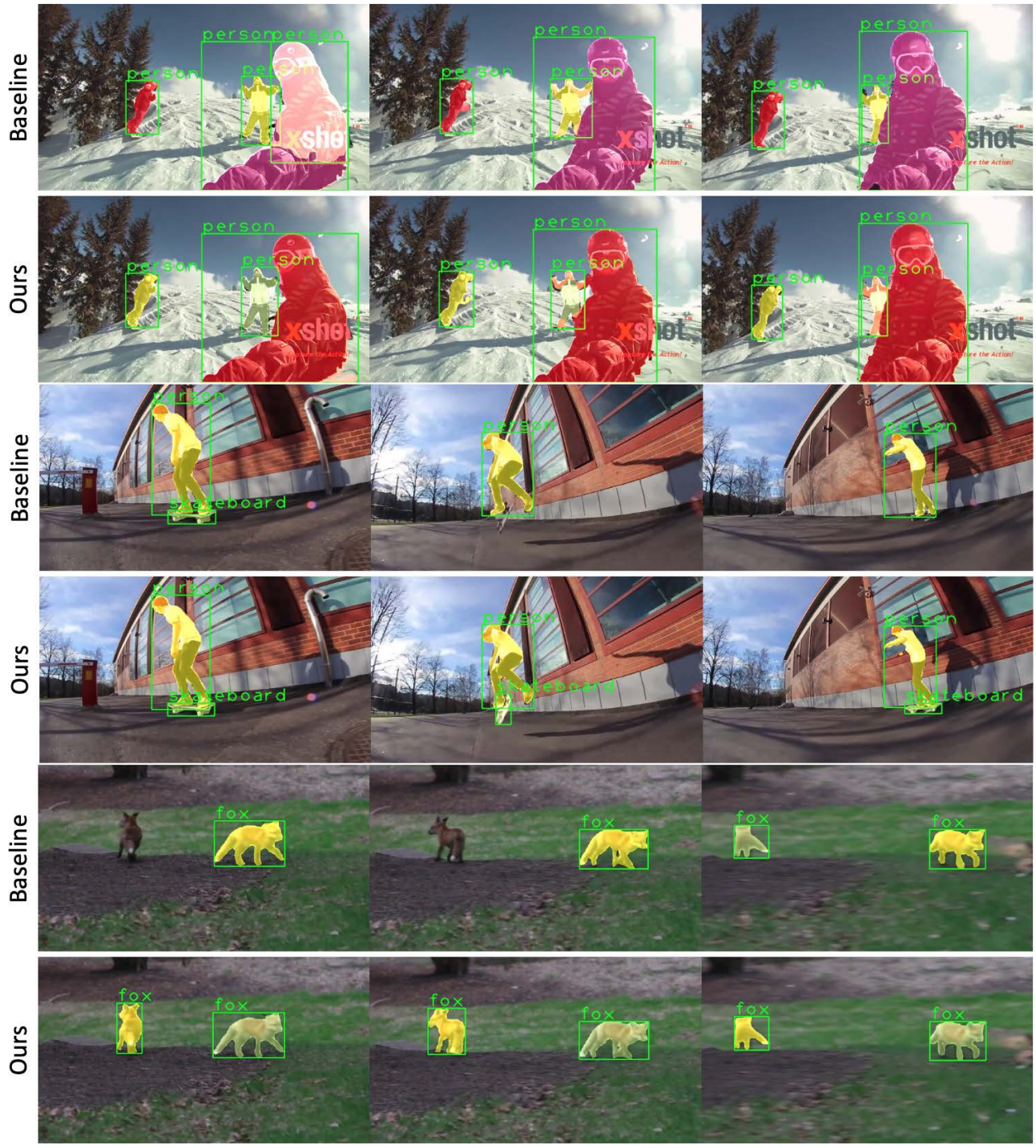


FIGURE 11. Example of visualized results between the baseline (MaskTrack R-CNN [57]) and Ours on YouTube-VIS `val`. Results are plotted if their confidence score is larger than 0.45. We confirm that the proposed method suppresses false negative detections. Best viewed digitally and in color.

To apply the proposed method to VIS, we replaced Faster R-CNN with MaskTrack R-CNN [57], an extension of Mask R-CNN [58] with a tracking branch to link the same object instances across two frames.

We compared our methods with TF-Blender [49] and TROI [48], which refined features to improve accuracy. The results with ResNet-50 are shown in Table 15. Our proposed method outperforms them on all evaluation metrics. With our proposed method, MaskTrack R-CNN is improved by more than 8.7% on the AP metric. TF-Blender utilizes nearby frames, but it aggregates frame-wise features, and the gain is limited. This is because instance segmentation requires more precise feature refinement. TROI proposes a temporal ROI alignment to extract ROI features from other frames based on their similarity; however, it is not sufficient for hard-to-detect scenes because the refinement is for the object-level features.

TABLE 15. Performance comparison with the state-of-the-art models on YouTube-VIS2019 `val`. All the methods use ResNet-50 as the backbone.

Methods	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack R-CNN [57]	30.3	51.1	32.6	31.0	35.5
MaskTrack R-CNN [57] + TF-Blender [49]	31.4	52.3	33.5	31.9	36.5
MaskTrack R-CNN [57] + TROI [48]	33.5	57.0	36.6	–	–
MaskTrack R-CNN [57] + **VSTAM(Ours)**	39.0	61.2	42.9	38.9	47.6

On the contrary, our approach is based on element-wise aggregation before object candidate detection, allowing us to improve the representation more precisely.

Figure 11 shows VIS results between the baseline (MaskTrack R-CNN [57]) and the proposed method on example frames in the validation set. We see that by refining element-wise features with the temporal information, false negatives are reduced, and masks are stabilized.

VI. CONCLUSION

We introduced a novel framework, VSTAM, for video object detection. It element-wisely refines features spatiotemporally, considering object misalignment before detection. The proposed video sparse transformer in VSTAM efficiently aggregates features sparsely with considerable time and memory cost. Moreover, we demonstrated significant accuracy improvements by storing the most utilized frame features during the aggregation in external memory. Extensive evaluations also demonstrated that it outperforms SOTAs on publicly available datasets.

The detailed error analysis reveals that our method significantly reduces background false positive and false negative detection. To achieve more stable detection, reducing class misclassification is necessary. We plan to incorporate tracking for feature refinement with continuing to detect the same object in the same class over time.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[2] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6569–6578.

[3] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 408–417.

[4] H. Wu, Y. Chen, N. Wang, and Z.-X. Zhang, "Sequence level semantics aggregation for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9217–9225.

[5] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10337–10346.

[6] S. Wang, Y. Zhou, J. Yan, and Z. Deng, "Fully motion-aware network for video object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 542–557.

[7] X. Chen, J. Yu, and Z. Wu, "Temporally identity-aware SSD with attentional LSTM," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2674–2686, Jun. 2020.

[8] M. Zhu and M. Liu, "Mobile video object detection with temporally-aware feature maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5686–5695.

[9] C. Zhang and J. Kim, "Video object detection with two-path convolutional LSTM pyramid," *IEEE Access*, vol. 8, pp. 151681–151691, 2020.

[10] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Relation distillation networks for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7023–7032.

[11] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6678–6687.

[12] M. Fujitake and A. Sugimoto, "Temporal feature enhancement network with external memory for object detection in surveillance video," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 7684–7691.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, C. A. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[14] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Comput. Vis. Image Understand.*, vol. 193, Apr. 2020, Art. no. 102907.

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[17] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3057–3065.

[18] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, and X. Wang, "Object detection in videos with tubelet proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 889–897.

[19] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, and W. Ouyang, "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2896–2907, Oct. 2018.

[20] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and S. T. Huang, "Seq-NMS for video object detection," in *Proc. Workshops Int. Conf. Learn. Represent.*, 2016, pp. 1–4.

[21] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7210–7218.

[22] J. Kim, J. Koh, B. Lee, S. Yang, and J. W. Choi, "Video object detection using object's motion context and spatio-temporal feature aggregation," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 1604–1610.

[23] C. Guo, B. Fan, J. Gu, Q. Zhang, S. Xiang, V. Prinet, and C. Pan, "Progressive sparse local attention for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3909–3918.

[24] 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 *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 18–34.

[25] L. Lin, H. Chen, H. Zhang, J. Liang, Y. Li, Y. Shan, and H. Wang, "Dual semantic fusion network for video object detection," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1855–1863.

[26] L. Han, P. Wang, Z. Yin, F. Wang, and H. Li, "Exploiting better feature aggregation for video object detection," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1469–1477.

[27] M. Han, Y. Wang, X. Chang, and Y. Qiao, "Mining inter-video proposal relations for video object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 431–446.

[28] G. Sun, Y. Hua, G. Hu, and N. Robertson, "MAMBA: Multi-level aggregation via memory bank for video object detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 2620–2627.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[30] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[31] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, and L. Yang, "Big bird: Transformers for longer sequences," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1–15.

[32] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," 2019, *arXiv:1904.10509*.

[33] G. Zhao, J. Lin, Z. Zhang, X. Ren, Q. Su, and X. Sun, "Explicit sparse transformer: Concentrated attention through explicit selection," 2019, *arXiv:1912.11637*.
[34] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 813–824.
[35] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "SSTVOS: Sparse spatiotemporal transformers for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5912–5921.
[36] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–11.
[37] Q. Zhou, X. Li, L. He, Y. Yang, G. Cheng, Y. Tong, L. Ma, and D. Tao, "TransVOD: End-to-end video object detection with spatial-temporal transformers," 2022, *arXiv:2201.05047*.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
[40] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–10.
[41] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2
[42] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–10.
[43] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 331–346.
[44] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*.
[45] K. He, R. Girshick, and P. Dollár, "Rethinking ImageNet pre-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4918–4927.
[46] M. Shvets, W. Liu, and A. Berg, "Leveraging long-range temporal relationships between proposals for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9756–9764.
[47] 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 *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1507–1516.
[48] 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 *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 2, pp. 1442–1450.
[49] Y. Cui, L. Yan, Z. Cao, and D. Liu, "TF-blender: Temporal feature blender for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8138–8147.
[50] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
[51] 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," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
[52] K.-J. Kim, P.-K. Kim, Y.-S. Chung, and D.-H. Choi, "Performance enhancement of YOLOv3 by adding prediction layers with spatial pyramid pooling for vehicle detection," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.
[53] K.-J. Kim, P.-K. Kim, Y.-S. Chung, and D.-H. Choi, "Multi-scale detector for accurate vehicle detection in traffic surveillance data," *IEEE Access*, vol. 7, pp. 78311–78319, 2019.
[54] H. Perreault, G.-A. Bilodeau, N. Saunier, and M. Heritier, "SpotNet: Self-attention multi-task network for object detection," in *Proc. 17th Conf. Comput. Robot Vis. (CRV)*, May 2020, pp. 230–237.
[55] H. Perreault, G.-A. Bilodeau, N. Saunier, and M. Héritier, "FFAVOD: Feature fusion architecture for video object detection," *Pattern Recognit. Lett.*, vol. 151, pp. 294–301, Nov. 2021.
[56] D. Bolya, S. Foley, J. Hays, and J. Hoffman, "TIDE: A general toolbox for identifying object detection errors," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 558–573.
[57] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5188–5197.
[58] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

MASATO FUJITAKE (Graduate Student Member, IEEE) received the master's degree in electrical, electronic and communications engineering technology from the Shibaura Institute of Technology, Tokyo, Japan, in 2019. He is currently pursuing the Ph.D. degree in computer science with SOKENDAI, Japan. His current research interests include deep learning and video analysis.

AKIHIRO SUGIMOTO (Member, IEEE) received the B.S., M.S., and Dr. Eng. degrees in mathematical engineering from The University of Tokyo, in 1987, 1989, and 1996, respectively.

He joined the Hitachi Advanced Research Laboratory, in 1989, and then temporally moved to the Advanced Telecommunications Research Institute International (ATR), Japan, in 1991. In 1995, he returned to the Hitachi Advanced Research Laboratory, where he lead a project on content-based image retrieval supported by the Ministry of International Trade and Industry in Japan. In 1999, he moved to Kyoto University as a Lecturer with the Graduate School of Informatics. Since 2002, he has been working with NII, where he is currently a Full Professor. From 2006 to 2007, he was also a Visiting Professor with the University of Paris-Est Marne-la-Vallée, France. He has published more than 150 peer-reviewed journals/international conference papers. He received the Best Paper Awards from the Information Processing Society of Japan, in 2001, and the Institute of Electronics, Information and Communication Engineers (IEICE), in 2011. He is a member of ACM. He is a regular reviewer of international conferences/journals in computer vision, AI, and pattern recognition. He has been an Associate Editor of *International Journal of Computer Vision*, since 2014. He also served several top-tier conferences, including IEEE/CVF International Conference on Computer Vision (ICCV), European Conference on Computer Vision (ECCV), Asian Conference on Computer Vision (ACCV), International Conference on Pattern Recognition (ICPR), and International Conference of 3D Vision (3DV) as Area Chair, the Program Chair, and the General Chair.