# Delving Deeper Into Mask Utilization in Video Object Segmentation

Mengmeng Wang, Jianbiao Mei, Lina Liu, Guanzhong Tian, *Member, IEEE*, Yong Liu, *Member, IEEE*, and Zaisheng Pan

***Abstract*— This paper focuses on the mask utilization of video object segmentation (VOS). The mask here mains the reference masks in the memory bank, i.e., several chosen high-quality predicted masks, which are usually used with the reference frames together. The reference masks depict the edge and contour features of the target object and indicate the boundary of the target against the background, while the reference frames contain the raw RGB information of the whole image. It is obvious that the reference masks could play a significant role in the VOS, but this is not well explored yet. To tackle this, we propose to investigate the mask advantages of both the encoder and the matcher. For the encoder, we provide a unified codebase to integrate and compare eight different mask-fused encoders. Half of them are inherited or summarized from existing methods, and the other half are devised by ourselves. We find the best configuration from our design and give valuable observations from the comparison. Then, we propose a new mask-enhanced matcher to reduce the background distraction and enhance the locality of the matching process. Combining the mask-fused encoder, mask-enhanced matcher and a standard decoder, we formulate a new architecture named MaskVOS, which sufficiently exploits the mask benefits for VOS. Qualitative and quantitative results demonstrate the effectiveness of our method. We hope our exploration could raise the attention of mask utilization in VOS.**

***Index Terms*— Video object segmentation, reference mask utilization, multi-scale mask fusion, mask-enhanced matcher.**

## I. Introduction

IN RECENT years, Video Object Segmentation (VOS) has received great attention due to its wide applications like video manipulation and editing. This paper focuses on the semi-supervised video object segmentation task, which segments target objects over video sequences with only an initial mask given. This technique has dramatically simplified video manipulation and editing applications by enabling users to merely segment the target objects on the first frame then the targets in the following frames will be segmented automatically, as opposed to process the whole video manually and painstakingly.

There are two kinds of reference information in VOS, i.e., reference frames and corresponding reference masks. The former is some raw RGB images of the previous video sequence, which have been processed and segmented. It contains the raw and whole information of the object as well as the background. The latter is the corresponding predicted masks of the frames (given mask template for the first frame). It depicts the edge and contour features of the object and explicitly indicates the boundary of the target against the background. The two kinds of information are used for memorizing the historical target information and current target feature matching. Although the reference mask is helpful for accurate segmentation in VOS, we note that it remains an open problem of how to appropriately make use of it and efficiently fuse it with the frames for better target memorizing and matching. Most previous methods [1], [6], [7], [8], [9], [10], [11] consider some naive ways, which only treat masks as simple auxiliary and pay less attention to further mining the features in masks and effective fusion with frame features. For example, MaskTrack [6], RGMP [7] and STM [1] simply concatenate the frame and the mask as the input of the network. The use of reference masks in these methods is insufficient since they do not dig deep into the mask representation, its combination with frame features, and its influence on the matcher, which could benefit the segmentation quality.

Until recently, the reference mask utilization in VOS has aroused some attention [5], [12], [13]. For example, SwiftNet [12] generates mask features via convolutions and revered sub-pixel modules for efficient reference encoding. However, besides mask usage, these methods always have other specific designs and different experimental settings to improve their performance, like the network architectures, training and inference configurations, hyper-parameters, other special modules, etc. It is hard to figure out the most effective manner of using the reference masks among these methods and if there is a better way. Moreover, previous research on mask employment mainly focuses on the feature embedding part but ignores the memory retrieval process, which is also crucial to the segmentation results.

In this paper, we delve deeper into reference mask utilization in VOS. First, we list eight different encoders to find a sufficient way of employing the reference masks in the

Manuscript received 28 February 2022; revised 22 August 2022; accepted 10 September 2022. Date of publication 27 September 2022; date of current version 3 October 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61836015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yulan Guo. *(Mengmeng Wang and Jianbiao Mei contributed equally to this work.) (Corresponding author: Yong Liu.)*

Mengmeng Wang, Jianbiao Mei, Lina Liu, Yong Liu, and Zaisheng Pan are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: mengmengwang@zju.edu.cn; jianbiaomei@zju.edu.cn; linaliu@zju.edu.cn; yongliu@iipc.zju.edu.cn; panzs@zju.edu.cn).

Guanzhong Tian is with the Ningbo Research Institute, Zhejiang University, Ningbo 315000, China (e-mail: gztian@zju.edu.cn).

Digital Object Identifier 10.1109/TIP.2022.3208409


Fig. 1. Comparison of the attention map between an ordinary matcher [1], [2], [3], [4], [5] and our mask-enhanced matcher. We first compute the similarity scores for the pixels inside the ground-truth object area of the query image, then visualize the normalized soft weights to the first and previous frames in the first two columns. The third column shows final mask predictions. Some details are zoomed in with red rectangles. The proposed mask-enhanced matcher could concentrate more on the target regions and suppress the background distraction.

encoder part. Half of the eight instantiations are inherited or summarized from existing methods, and the other half types are designed by ourselves. Next, to benchmark their effectiveness, we provide a unified codebase that implements all the eight instantiations and keeps the same remaining architectures (matcher and decoder), training/inference configurations and hyper-parameters for every encoder. We will open-source the codebase to the research community for easy model re-implementation, analysis and comparison. One of our designs achieves the best results among the eight instantiations and is regarded as our final encoder choice. We empirically conclude two key findings from the comparison: (i) independent mask representation with a separate encoder is necessary, which is more beneficial than using raw masks or simple downsampling, and (ii) multi-scale fusion of the mask features and frame features improves the performance, demonstrating that both the low-level and high-level mask features are useful.

Next, we try to explore the usage of the reference masks on the matcher. Though this is always omitted by previous methods, we find it is helpful to eliminate the background distraction. The conventional matcher in VOS is a non-local-like block [1] and lots of papers follow this way [1], [2], [3], [4], [5]. However, the attention map between query frame and reference sets in this kind of matcher concerns numerous unnecessary feature pairs (e.g., the relationships among the backgrounds), thus containing too much background noise and distractions. We believe this problem could be easily improved by explicitly introducing the reference mask information into the matcher. Unlike previous matchers, we propose a new mask-enhanced matcher, which first uses the mask features to generate a mask attention map, then applies the mask attention map on the value embeddings by Hadamard product to enhance the target region and suppress the backgrounds. As shown in Fig. 1, the proposed matcher could concentrate more on the target object and less on the backgrounds than the commonly-used matcher [1], [2], [3], [4], [5]. Now, combining the proposed mask-fused encoder, the mask-enhanced matcher and a conventional decoder, we formulate a new architecture, dubbed **MaskVOS**, which sufficiently exploits the advantages of the masks for VOS in both the encoder and matcher. We extensively evaluate our model on three representative and standard VOS datasets, namely DAVIS16 [14], DAVIS17 [15] and YouTube-VOS [16], showing its superior performance over most existing methods.

The main contributions of this paper could be summarized as follows:

- We provide a unified testbed for eight different VOS encoders to investigate an effective mask fusion strategy. This is the first work to compare a wide range of VOS models from the perspective of mask utilization under the same experimental conditions.
- We explore the mask benefits on the matcher and propose an insightful mask-enhanced matcher to eliminate the background distraction and enhance the target features in the matching process.
- We propose a new network, dubbed MaskVOS, which sufficiently makes use of the reference masks in both the encoder and matcher. The effectiveness of our model is demonstrated on three benchmark datasets, highlighting the importance of effectively using the mask in VOS.

## II. Related Works

### A. Online Learning Methods

Online-learning methods [4], [6], [17], [18], [19], [20], [21], [22] usually finetune the segmentation network by using the first frame and corresponding mask during inference to identify the appearance of the target object in the remaining video frames. These methods use online adaptation [17], instance segmentation information [18], data augmentation techniques [19], or an integration of multiple techniques [21]. OSVOS [20] is the first online approach to exploit deep learning for the VOS problem, where a multi-stage training strategy is designed to gradually shrink the focus of the network from general objects to the one in reference masks. PreMVOS [21] combined multiple techniques and adapted the network to the target video domain by finetuning on a large set of augmented images generated from the first-frame ground truth. FRTM [22] integrates a lightweight discriminative target model and a segmentation network for modeling the target appearance and generating accurate segmentation masks. STM-cycle [4] design a gradient correction module with a cyclic mechanism to extend the offline segmentation network to an online approach. While online learning can achieve high-quality segmentation and is robust against occlusions, it is computationally expensive as it requires finetuning for each video.

### B. Tracking-Based Methods

The Siamese network-based trackers have drawn great attention in recent years, which model tracking as a template matching task and perform similarity learning. By introducing

the powerful backbones [23], [24] and elaborated prediction networks [25], [26], [27], [28], Siamese trackers obtain superior performance and the Siamese-like feature extractor becomes a default configuration. The VOS task has some relationships with the visual tracking task, where the common characteristics are that (1) they both need to track specific targets and (2) they face similar challenges like similar distractors and occlusions. The most significant difference is that visual tracking only needs to output the target's bounding box location, while VOS needs to predict finer pixel-wise masks. There are some works [29], [30], [31] that realize the relationships between the two tasks and propose to unite them together, achieving fast inference speed. For example, SiamMask [29] improves the offline training procedure of the popular Siamese tracker SiamRPN [26] by augmenting their loss with a binary segmentation task, narrowing the gap between tracking and segmentation. SAT [30] fuses object tracking and segmentation into a unified pipeline. It combines SiamFC++ [27] and proposed an estimation-feedback mechanism to switch between mask box and tracking box, making segmentation and tracking tasks enhance each other. The integration of tracker help to improve the inference speed, while the accuracy of tracking often limits these methods' performance.

### C. Matching-Based Methods

To capture the information that lies in the historical frames, many methods [1], [2], [3], [11], [12], [32], [33], [34], [35] perform feature matching at the pixel level to learn target object appearances offline and achieve state-of-the-art performance. VideoMatch [32] takes the first frame as a reference set and measures similarity by soft matching with foreground and background features. After that, FEELVOS [11] and CFBI [36] perform the nearest neighbor matching between the current frame and the first and previous frames in the feature space and utilizes the output of feature matching as internal guidance of the network. RANet [10] proposes a ranking attention module to increase the usefulness of the generated similarity map. To take more temporal cues from all past frames, STM [1] introduces an external memory to store past frames' features and uses the attention-based matching method to retrieve information from memory. Many methods [2], [3], [5], [33], [34], [37], [38], [39] are extended from STM. For instance, to reduce the non-locality, KMN [37] applies Query-to-Memory matching with a kernelized memory read. RMNet [34] proposes to replace STM's global-to-global matching with local-to-local matching. SwiftNet [12] elaborately compresses spatiotemporal redundancy in matching-based VOS via Pixel-Adaptive Memory. Some methods [13], [40], [41], [42], [43], [44], [45], [46] use vision transformers to capture spatial-temporal dependencies among frames.

We also regard STM as the baseline. Different from the above methods, we tackle the VOS problem from the perspective of reference mask utilization. Most of the existing methods tend to fuse masks in some direct and naive manners in the encoder part, and we aim to discuss this problem more comprehensively.

## III. Method

In this section, we elaborate on the technical details of our approach. First, we briefly describe the common VOS pipeline for a better understanding. Then, we introduce different reference mask utilization strategies with eight encoders and give the key observations. Next, we present the proposed mask-enhanced matcher and the formulated network in detail.

### A. Overview of the Common VOS Pipeline

The semi-supervised VOS problem can be defined as: given a video sequence of and a target mask in the first frame, the task is to segment sequential frames at every timestamp according to the reference set which includes several historical frames with their corresponding predicted masks (groundtruth for the initial frame). A typical pipeline of the commonly-used matching-based segmentation usually consists of a query encoder, a reference encoder, a matcher and a decoder.

*1) Query Encoder:* The query encoder $\mathcal{E}_Q$ is used to encode the $t$-th query frame to be segmented. It first obtains the feature representation of the query frame and then embeds it to its corresponding key and value embeddings ($\mathbf{k}^Q \in \mathbb{R}^{H\times W\times C_k}$, $\mathbf{v}^Q \in \mathbb{R}^{H\times W\times C_v}$) through two parallel convolution layers, where $H$ and $W$ are feature height and width, and $C_k$, $C_v$ are channel dimensions of key and value embeddings, respectively.

*2) Reference Encoder:* The reference encoder $\mathcal{E}_R$ is designed to memorize the target appearances from reference sets $\mathbf{R} = \{(\mathbf{I}_i, \mathbf{M}_i)\}_N$, where $N$ is the size of the reference set, including some past frames with their corresponding masks. Similar to the query encoder, the reference encoder also outputs key and value embeddings. If there is more than one element in reference sets, each of them will be independently encoded. Mathematically, the final output of the reference encoder is a pair of key and value embeddings ($\mathbf{k}^R \in \mathbb{R}^{N\times H\times W\times C_k}$, $\mathbf{v}^R \in \mathbb{R}^{N\times H\times W\times C_v}$).

*3) Matcher:* The matcher $\mathcal{M}$ is applied to model the relationship between the query frame and the reference sets. It could also be illustrated as retrieving the target information from a memory bank (reference set). The matcher first calculates the similarity of every pixel in the query key embedding $\mathbf{k}^Q$ with every pixel in reference key embedding $\mathbf{k}^R$. Then, the similarity matrix is regarded as the affinity matrix to be multiplied on the reference value $\mathbf{v}^R$. Next, the results will be concatenated with the query value $\mathbf{v}^Q$ as the final matching results $\mathbf{y}$. The matching procedure can be expressed as:

$$\mathbf{y}(p) = [\mathbf{v}^Q(p), \sum_{\forall q} \sigma(\mathbf{k}^Q(p), \mathbf{k}^R(q)) \cdot \mathbf{v}^R(q)] \tag{1}$$

where $p$ and $q$ denote pixels in query and reference key embeddings, respectively. "$\cdot$" denotes dot-product and $\sigma$ is a softmax function $\sigma(k^Q(p), k^R(q)) = \frac{\exp(k^Q(p)\cdot k^R(q))}{\sum_{\forall q}\exp(k^Q(p)\cdot k^R(q))}$. For multi-objects segmentation, the above matching process will be implemented for every target in a parallel manner.

*4) Decoder:* The decoder $\mathcal{D}$ receives the output of the matcher as input, then output segmentation results for each target in the query frame. It includes a refinement module [7] that gradually upscales the feature maps with multiple stages.

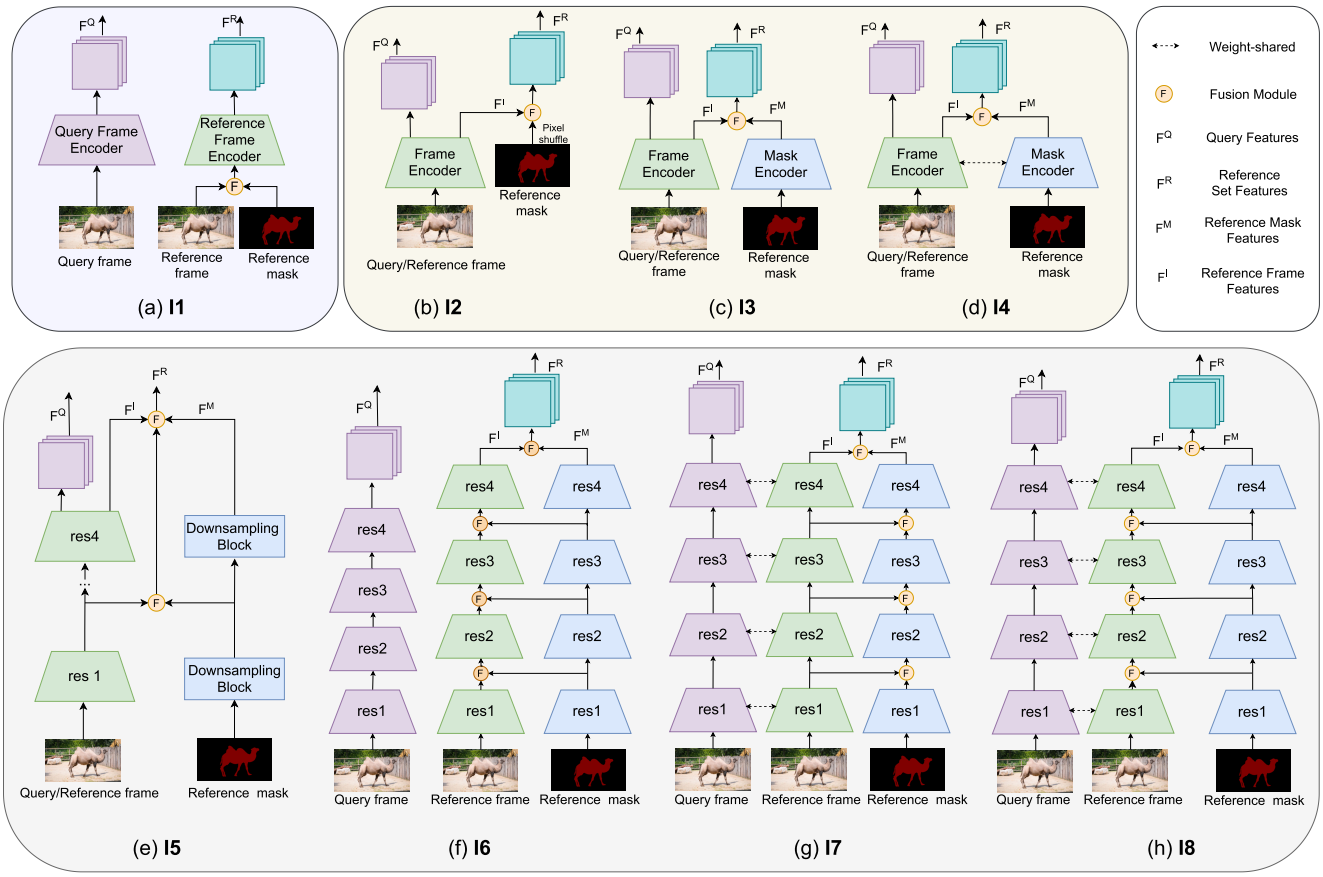


Fig. 2. Mask utilization on the encoder. Eight instantiations are implemented. The light purple, faint yellow and light grey rounded rectangles in the background correspond to early fusion, post fusion and multi-scale fusion, respectively.

At every stage, the output of the previous stage will be concatenated with a feature map from the query encoder at the corresponding scale through skip-connections and then fed into the next stage. A softmax operation is attached on the last refinement stage and the mask probabilities for each target in the current frame are acquired. Finally, a soft aggregation operation [7] is used for merging all predicted maps to obtain the segmentation results.

### *B. Reference Mask Utilization on the Encoder*

Reference masks are used by most prior methods [1], [6], [7], [8], [9], [10], [11], [36] in the encoder part to obtain fine-grained segmentation by default. However, most of the existing methods tend to absorb the reference masks in some direct and naive manners, without further exploration. In this section, we study the mask utilization in the encoder and try to find the most effective mask fusion way from eight different kinds of encoders. The encoders are classified into three categories as early fusion, post fusion and multi-scale fusion, according to the mask fusion location in the encoders, as shown in Fig. 2. Half of the eight instantiations are inherited or summarized from existing methods and the other half are devised by ourselves. Note that the instantiations from the previous approaches might not cover all the possible cases but select several representative methods. Next, to make the comparison about different mask combination ways as fair as possible, we provide a unified codebase that keeps the same remaining architectures (matcher and decoder [1]), training/inference configurations and hyper-parameters for every instantiation. Details are as follows.

**Early fusion** is the most naive and frequently-used way, which fuses the raw reference masks with reference frames before feeding into the reference encoder. The majority of previous methods [1], [2], [6], [7], [9], [33], [35], [37], [39], [47] lie in this kind. For example, the most representative method STM [1] and STM-based methods [34], [37], [39] exploit two encoders, i.e. a 4-channel memory encoder for the reference frames with the predicted masks encoding and a regular query encoder for the query frame feature extraction. We realize one instantiation (listed as I1) from these STM-based methods as shown in Fig. 2 (a). It simply concatenates the reference frames with corresponding raw masks along the channel dimension to form a 4-channel input tensor and then sends it into a reference encoder. Specifically,

$$\mathbf{R} = \{Concate(\mathbf{I}_i, \mathbf{M}_i)\}_N \tag{2}$$

where $\mathbf{I}_i$ and $\mathbf{M}_i$ denotes the frame and mask (groundtruth for the first frame) of the $i$-th sample in the reference set $\mathbf{R}$, respectively. $N$ is the size of $\mathbf{R}$ and $Concate(\cdot)$ indicates the concatenation operation along the channel dimension. The mask is regarded as a channel of the frame thus only has the same importance as the three (R, G, B) color channels. There is no further explicit spatial interaction between frames

and masks, but the channel fusion is considered inside the encoder.

**Post fusion** aggregates the raw mask or mask features with the frame features after the feature extraction stage for foreground discovery [5], [8], [9], [10], [11], [34], [36], [40], [48], [49]. For instance, AGAME [8] concatenates the raw masks with the frame embeddings to enhance the target appearances. RaNet [10] utilizes the raw mask to select foreground (FG) or background (BG) similarity maps as FG or BG features for segmentation. SSTVOS [40] uses the mask to get object affinity value from the transformer for predicting object masks. Formally, in post fusion, we have

$$\mathbf{F}^R = f(\mathbf{F}^I, \mathbf{F}^M) \tag{3}$$

where $\mathbf{F}^R$, $\mathbf{F}^I$ and $\mathbf{F}^M$ denote the features of the reference set, reference frames and reference masks, respectively. $f$ means the fusion function and follows [7] for all encoders in this type. We have summarized two common mask fusion approaches from exiting methods in this type and implemented their mask utilization part with two corresponding instantiations, I2 [8], [9], [10] and I3 [5]. In detail, I2 (Fig. 2(b)) directly downsamples the raw masks and fuses them with output reference frame features from the query/reference shared frame encoder. Differently, I3 (Fig. 2(c)) proposes to extract the mask features with an independent network. Besides, we add I4 (Fig. 2(d)) to explore if the mask encoder could be shared with the frame encoder. Post fusion combines the high-level features of reference frames and masks, and no low-level interaction is considered.

**Multi-scale fusion** is exploited very recently [12] and means using mask information in multiple locations of the encoders. Specifically, SwiftNet [12] uses multi-stage downsampling blocks for mask embedding and fusion. We implement one instantiation (I5) from the mask fusion part of SwiftNet and propose three other instantiations (I6, I7 and I8) for a more in-depth discussion. I5 (Fig. 2(e)) fuses downsampled mask information into the frame features after the first and fourth ResNet stages. The frame features and mask features are aligned vertically with the same size and concatenated together to facilitate multi-scale aggregation. Besides, the query and reference frames share one encoder. The fusion module (elementwise sum) and downsampling blocks (reversed sub-pixel modules and $1\times1$ convolutions) of I5 are consistent with SwiftNet. I6 (Fig. 2(f)) employs a separate ResNet to extract the reference mask features rather than simple downsampling, then fuses them with the reference frame features in the first four stages of ResNet. The query and reference frame encoders are not shared. I7 (Fig. 2(g)) differs from I6 in two folds, (1) the query and reference frame encoders are shared; (2) the reference mask encoder is regarded as the mainstream so that the features of reference frames could be reused. Therefore, in I7, every frame only needs to forward once through the frame encoder and could be directly added into the reference set if necessary. For I8 (Fig. 2(h)), the difference with I6 is that a shared frame encoder is used for the query and reference frames to reduce the parameters. The fusion modules in I6, I7 and I8 are the same, which is an AFC block [50].

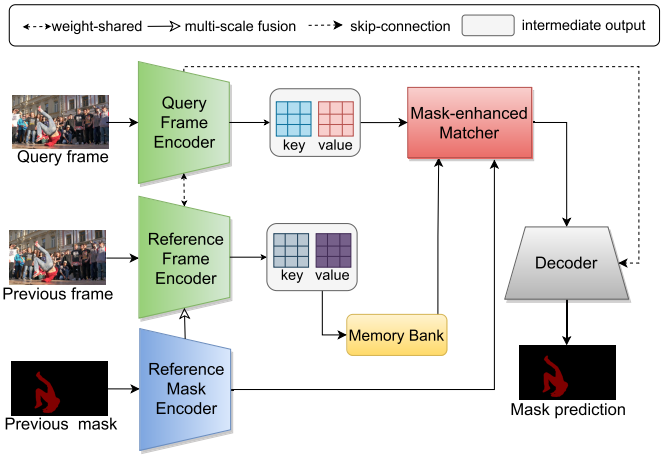


Fig. 3. Detailed pipeline of the proposed MaskVOS. It consists of an unbalanced Siamese frame encoder for query and reference frame feature extraction, a mask encoder for mask representation, a memory bank to save the reference set, a mask-enhanced matcher to retrieve the memory bank and a decoder to obtain the final segmentation results.

Multi-scale fusion of masks provides multi-level interactions, enabling sufficient fusion for reference masks and frames.

From the comparison of the above eight instantiations in Sec. IV-E, we observe that our I8 obtains the best performance with a shared frame encoder, a separate mask encoder and the multi-scale fusion. Our empirical evaluation results reveal that (i) extracting mask representations with a stand-alone feature encoder is important, which is more helpful than using raw masks or simple downsampling, and (ii) multi-scale fusion is the most sufficient manner and achieves the best results, demonstrating both the low-level and high-level mask features are useful.

### C. MaskVOS

According to the above analyses of the mask influence in the encoder part, a problem is raised, "*Could the reference masks also benefit the matcher?*" To investigate this, we propose a mask-enhanced matcher, exploiting the mask features during the memory matching process to further enhance target features and weaken the distraction of background areas. Then, by combining the encoder of I8 from Sec. III-B, the proposed mask-enhanced matcher and an ordinary decoder from Sec. III-A, we propose a new method, dubbed **MaskVOS**, which thoroughly excavates the mask in both the encoder and matcher parts of VOS. The pipeline of MaskVOS is presented in Fig. 3.

We first give more details about our encoder here. As shown in Fig. 2(h) and Fig. 3, the proposed feature extractor consists of an unbalanced Siamese frame encoder and a light mask encoder. The former extracts features of query and reference frames, while the latter takes the reference masks as input and outputs multi-scale mask features. The Siamese frame encoder is "unbalanced" since it encodes the query and reference frames differently. When extracting the features of the query frame, the Siamese encoder maps the RGB frame to the embedding space directly. When encoding the reference frame,

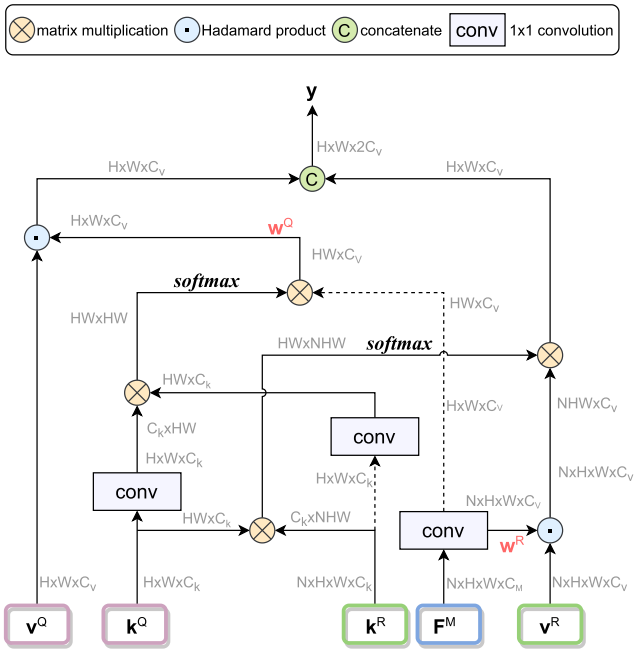


Fig. 4. Detailed architecture of the proposed mask-enhanced matcher as described in Sec. III-C. The dashed arrow means slicing the last element along the temporal dimension. The $\mathbf{w}^R$ and $\mathbf{w}^Q$ are the mask attention maps of the reference sets and the query frame.

it fuses the mask features from the mask encoder through the AFC [50] modules in multiple feature scales.

Most prior arts [2], [3], [4], [5], [39] use the non-local like matcher [1], but the VOS is actually a local task since every pixel on the target object mainly has relationships with neighboring pixels. The attention map between the query frame and reference sets in this kind of matcher includes plenty of unnecessary feature pairs, thus containing background noise and not so focused on the target, as shown in Fig. 1. One of the solutions is to strengthen the object features and suppress the backgrounds by directly involving the reference mask information in the matcher. However, this is always neglected by the previous works. Instead, we realize this situation and develop a mask-enhanced matcher, which utilizes the reference mask features produced by the light mask encoder to reduce the non-localization. As presented in Fig. 4, we first use the mask features $\mathbf{F}^M$ to generate the mask attention maps. The attention maps are multiplied with the value embeddings, obtaining the mask-enhanced values. More specifically, for reference sets, we employ a $1 \times 1$ convolution layer to $\mathbf{F}^M$ to generate the mask attention map $\mathbf{w}^R$ directly, and obtain the aggregated value $\hat{\mathbf{v}}^R$ as:

$$\begin{aligned} \mathbf{w}^R &= Conv(\mathbf{F}^M) \\ \hat{\mathbf{v}}^R &= \mathbf{w}^R \circ \mathbf{v}^R \end{aligned} \tag{4}$$

where $Conv$ denotes $1 \times 1$ convolutions. For the attention map $\mathbf{w}^Q$ of the query frame, we need to transfer the previous mask features $\mathbf{F}^M$ to the current frame since the mask features of the current frame is not available yet. Instead of intuitive time-consuming warping with optical flow, we simply translate the attention map of the last frame $\mathbf{w}^R_{-1}$ to the current frame according to the feature similarities between the query frame and previous frame (the $t$-1 frame) from the reference set as:

$$\begin{aligned} \mathbf{w}^Q &= \sigma(Conv(\mathbf{k}^Q), Conv(\mathbf{k}^R_{-1})) \times \mathbf{w}^R_{-1} \\ \hat{\mathbf{v}}^Q &= \mathbf{w}^Q \circ \mathbf{v}^Q \end{aligned} \tag{5}$$

where the subscript $-1$ denotes the last element of the reference set, i.e., the $t$-1 one, note that the ($t$-1)-th frame and corresponding mask are always stored in the reference set. $\sigma$ is the softmax function. $\circ$ and $\times$ denote Hadamard product and matrix multiplication, respectively. After the value enhancement, the final matching results of the matcher is:

$$\mathbf{y}(p) = [\hat{\mathbf{v}}^Q(p), \sum_{\forall q} \sigma(\mathbf{k}^Q(p), \mathbf{k}^R(q)) \cdot \hat{\mathbf{v}}^R(q)] \tag{6}$$

where $p$ and $q$ denote pixel in query and reference key embedding, respectively. $\mathbf{y}$ is the output of the matcher. Next, $\mathbf{y}$ will be sent to a decoder which is a common one as described in Sec. III-A, to obtain the final segmentation results.

## IV. Experiments

In this section, we first introduce the implementation details of our approach and the datasets and the evaluation metrics. Then we perform extensive experiments to demonstrate that the proposed MaskVOS consistently outperforms or obtains a comparable performance with the state-of-the-art methods on three datasets, DAVIS16 [14], DAVIS17 [15] and YouTube-VOS [16]. Next, we give some qualitative results to show the effectiveness of our MaskVOS. Finally, we conduct comprehensive ablation studies to analyze the effect of the individual components of our method and some configurations.

### A. Implementation Details

We implement two versions of our approach. The first uses a ResNet-50 [51] (the first four stages) as the backbone of the frame encoder, and a ResNet-18 (the first four stages) as the backbone of the mask encoder, except that of I4 is a ResNet-50 since it is shared with the frame encoder. The other one (also the final version) employs a swin-transformer-small [52] as the frame encoder's backbone and a swin-transformer-tiny as the mask encoder's backbone. The ResNet and transformer versions are shorted as MaskVOS† and MaskVOS, respectively. All the ResNet and swin-transformer models are pretrained on ImageNet [53]. Similar to most STM-based methods [1], [33], [38], we synthesize video clips with length 3 by applying data augmentations (random affine, color, flip, resize and crop) on COCO [53] training sets. Then we use these synthetic videos to pretrain our model. Adam optimizer [54] with a fixed learning rate of 1e-5 is used for pretraining. After that, we train our model on real videos by sampling 3 frames from each video sequence and applying data augmentation on those frames. The maximum time interval of sampling increases by 5 for every ten training epochs. We freeze all batch normalization layers and use AdamW optimizer ($\beta = (0.9, 0.999)$, $eps = 10^{-8}$) with an initial learning rate $4 \times 10^{-6}$. The model is trained with a batch size of 4 and the input resolution $400 \times 400$ for 160 epochs on 4 TITAN RTX GPUs. In the inference stage, our model takes the 480p

TABLE I
COMPARISON WITH THE STATE-OF-THE-ART ON THE DAVIS16-VAL AND DAVIS17-VAL. 'OL' INDICATES THE USE OF THE ONLINE-LEARNING STRATEGY. '+YT' MEANS THE USE OF YOUTUBE-VOS FOR TRAINING. THE RUNTIME OF OTHER METHODS WAS OBTAINED FROM THE CORRESPONDING PAPERS

| Methods | OL | DAVIS16 val | | | | DAVIS17 val | | |
|---|---|---|---|---|---|---|---|---|
| | | FPS | $\mathcal{J}\&\mathcal{F}(\%)$ | $\mathcal{J}(\%)$ | $\mathcal{F}(\%)$ | $\mathcal{J}\&\mathcal{F}(\%)$ | $\mathcal{J}(\%)$ | $\mathcal{F}(\%)$ |
| OSVOS [20] | ✓ | 0.22 | 80.2 | 79.8 | 80.6 | 60.3 | 56.7 | 63.9 |
| OnAVOS [17] | ✓ | 0.08 | 85.5 | 86.1 | 84.9 | 67.9 | 64.5 | 71.2 |
| PReMVOS [21] | ✓ | 0.03 | 86.8 | 84.9 | 88.6 | 77.8 | 73.9 | 81.7 |
| STM-cycle(+YT) [4] | ✓ | - | - | - | - | 72.3 | 69.3 | 75.3 |
| FRTM(+YT) [22] | ✓ | 21.9 | 83.5 | - | - | 76.7 | - | - |
| RGMP [7] | | 7.7 | 81.8 | 81.5 | 82.0 | 66.7 | 64.8 | 68.6 |
| RaNet [10] | | 30 | 85.5 | 85.5 | 85.4 | 65.7 | 63.2 | 68.2 |
| AGSS [9] | | - | - | - | - | 66.6 | 63.4 | 69.8 |
| GC [38] | | 25 | 86.6 | 87.6 | 85.7 | 71.4 | 69.3 | 73.5 |
| AFB-URR [2] | | - | - | - | - | 74.6 | 73.0 | 76.1 |
| AGAME(+YT) [8] | | 14.3 | 82.1 | 82.0 | 82.2 | 70.0 | 67.2 | 72.7 |
| FEELVOS(+YT) [11] | | 2.2 | 81.7 | 81.1 | 82.2 | 71.5 | 69.1 | 74.0 |
| STM(+YT) [1] | | 6.3 | 89.3 | 88.7 | 89.9 | 81.8 | 79.2 | 84.3 |
| KMN(+YT) [37] | | 8.3 | 90.5 | 89.5 | 91.5 | 82.8 | 80.0 | 85.6 |
| LCM(+YT) [39] | | 8.6 | 90.7 | 89.9 | 91.4 | 83.5 | 80.5 | 86.5 |
| STCN(+YT) [3] | | - | - | - | - | 85.4 | **82.2** | 88.6 |
| AOT-L(+YT) [13] | | 18.7 | 91.0 | 89.7 | **92.3** | 83.0 | 80.3 | 85.7 |
| EGMN(+YT) [33] | | - | - | - | - | 82.8 | 80.2 | 85.2 |
| CFBI(+YT) [36] | | 6 | 89.4 | 88.3 | 90.5 | 81.9 | 79.1 | 84.6 |
| JOINT(+YT) [41] | | - | - | - | - | 83.5 | 80.8 | 86.2 |
| SST(+YT) [40] | | - | - | - | - | 82.5 | 79.9 | 85.1 |
| RMNet(+YT) [34] | | 12 | 88.8 | 88.9 | 88.7 | 83.5 | 81.0 | 86.0 |
| GIEL(+YT) [5] | | - | - | - | - | 82.7 | 80.2 | 85.3 |
| SwiftNet(+YT) [12] | | 25 | 90.4 | **90.5** | 90.3 | 81.1 | 78.3 | 83.9 |
| MaskVOS†(+YT) | | 15.1 | 90.0 | 89.2 | 90.8 | 83.3 | 80.0 | 86.7 |
| MaskVOS(+YT) | | 9.2 | **91.1** | 89.9 | **92.3** | **85.5** | 82.0 | **89.0** |

(480 × 854) resolution as input like other methods and adds a frame and its predicted mask into the memory bank every five frames. We conduct all inference experiments on a single TITAN RTX GPU.

### B. Datasets and Evaluation Metrics

We evaluate our approach on DAVIS16 [14], DAVIS17 [15] and YouTube-VOS [16] benchmarks. DAVIS2016 is a single-object dataset, which contains 30 training video sequences and 20 validation video sequences. DAVIS2017 is a multi-objects dataset expanded from DAVIS2016, including 60 training video sequences and 30 validation video sequences. YouTube-VOS is a large-scale VOS dataset with 3471 training videos and 474 validation videos. And each video contains a maximum of 12 objects. The validation set includes seen objects from 65 training categories and unseen objects from 26 categories, appropriate for evaluating algorithms' generalization performance. We use the evaluation metrics provided by the DAVIS benchmark to evaluate our model. $J\&F$ evaluates the general quality of the segmentation results, $J$ evaluates the mask $IoU$ and $F$ estimates the quality of contours. We mainly use $J\&F$ to explain the performance in the following experiments.

### C. Comparison With the State-of-the-Art

*1) DAVIS:* Table I reports the evaluation results on DAVIS16-val and DAVIS17-val. Our MaskVOS achieves the leading performance on both DAVIS16-val (91.1%) and DAVIS17-val (85.5%), outperforming all the methods in the Table I. Specifically, it surpasses all the online-learning methods with a large margin (+4.3%∼+10.9%), the matching-based methods such as STM [1], RMNet [34] and CFBI [36], and the transformer-based methods SST [40] and JOINT [41]. Compared to the baseline STM, MaskVOS improves +1.8% on DAVIS16-val and +3.7% on DAVIS17-val. Moreover, our ResNet-based MaskVOS† also improves the baseline and already performers better than most methods like GIEL [5], SwiftNet [12], SST [40], EGMN [33] and so on in DAVIS17-val, and STM [1], RMNet [34], CFBI [36], etc in DAVIS16-val. We attribute the performance advances to the help of the proposed encoder and mask-enhanced matcher, which can effectively reduce background interference and enhance the target regions, thus greatly boost the quality of segmentation mask. On DAVIS17, with a ResNet-50 as the backbone, our method achieves 11.1 FPS (80.6%). Besides, compared with other transformer-based methods AOT-L [13], SST [40] and JOINT [41], our method achieves better results

TABLE II

COMPARISON WITH THE STATE-OF-THE-ART ON YOUTUBE-VOS 2018 VALIDATION SET. 'OL' INDICATES THE USE OF ONLINE-LEARNING STRATEGY. THE SUBSCRIPTS OF $\mathcal{J}$ AND $\mathcal{F}$ DENOTE SEEN OBJECTS ($s$) AND UNSEEN OBJECTS ($u$). THE METRIC OVERALL MEANS THE AVERAGE OF $\mathcal{J}_s, \mathcal{J}_u, \mathcal{F}_s, \mathcal{F}_u$

| Methods | OL | YouTube-VOS 2018 val | | | | |
|---|---|---|---|---|---|---|
| | | Overall | $\mathcal{J}_s(\%)$ | $\mathcal{J}_u(\%)$ | $\mathcal{F}_s(\%)$ | $\mathcal{F}_u(\%)$ |
| OSVOS [20] | ✓ | 58.8 | 59.8 | 54.2 | 60.5 | 60.7 |
| OnAVOS [17] | ✓ | 55.2 | 60.1 | 46.6 | 62.7 | 51.4 |
| STM-cycle [4] | ✓ | 70.8 | 72.2 | 62.8 | 76.3 | 71.9 |
| FRTM [22] | ✓ | 72.1 | 72.3 | 65.9 | 76.2 | 74.1 |
| RGMP [7] | | 53.8 | 59.5 | 45.2 | - | - |
| AGSS [9] | | 71.3 | 71.3 | 65.5 | 75.2 | 73.1 |
| AGAME [8] | | 66.1 | 67.8 | 60.8 | - | - |
| STM [1] | | 79.4 | 79.7 | 72.8 | 84.2 | 80.9 |
| GC [38] | | 73.2 | 72.6 | 68.9 | 75.6 | 75.7 |
| CFBI [36] | | 81.4 | 81.1 | 75.3 | 85.8 | 83.4 |
| AFB-URR [2] | | 79.6 | 78.8 | 74.1 | 83.1 | 82.6 |
| KMN [37] | | 81.4 | 81.4 | 75.3 | 85.6 | 83.3 |
| EGMN [33] | | 80.2 | 80.7 | 74.0 | 85.1 | 80.9 |
| SST [40] | | 81.7 | 81.2 | **76.0** | - | - |
| RMNet [34] | | 81.5 | **82.1** | 75.7 | 85.7 | 82.4 |
| GIEL [5] | | 80.6 | 80.7 | 75.0 | 85.0 | 81.9 |
| SwiftNet [12] | | 77.8 | 77.8 | 72.3 | 81.8 | 79.5 |
| MaskVOS† | | 81.5 | 80.5 | **76.0** | 85.2 | **84.3** |
| MaskVOS | | **81.9** | 81.4 | 75.9 | **86.6** | 83.9 |

than these methods due to the thorough mask utilization. Our runtime is a trade-off between latency and accuracy depending on requirement: if we reduce the input size to half of the input resolution, it achieves 28.1 FPS (71.1%).

*2) YouTube-VOS:* Table II shows the comparison with state-of-the-art methods on YouTube-VOS 2018 validation [16]. On this benchmark, our MaskVOS† obtains an overall score of 81.5%, significantly outperforming many state-of-the-arts models like STM [1], GIEL [5], KMN [37], EGMN [33] and SwiftNet [12]. Compared with the baseline STM, MaskVOS† boosts the performance with a dramatic gain +2.1%. Note that YouTube-VOS is a really large VOS dataset, thus +2.1% is actually a significant improvement. By using swin-transfomer as the backbone, MaskVOS further improves the performance to 81.9%. It surpasses all the methods in the table. Moreover, our method performs favorably on both seen and unseen categories.

### D. Qualitative Results

We show the qualitative comparison of our MaskVOS and the baseline STM [1] in Fig. 5. Benefitting from sufficiently absorbing the reference masks into both the encoder and matcher, our method yields more precise segmentation compared to STM. In the first video, even STM can segment out the primary instances, they lose the label for the bike in the fifth image and predict a confusion label for the handlebar when only part of it is visible in the last image. In the second sequence, STM fails to segment out the skateboard under occlusions and fast motion. By comparison, our method can robustly capture the object and succeed in tracking and segmentation due to our effective mask utilization.

Fig. 6 shows more qualitative examples of our MaskVOS. We choose challenging videos from DAVIS17 validation sets and sample important frames (e.g. before and after occlusions, dramatic deformation and complex motion). As seen in the figure, our method is robust to these challenges. For example, the bicycle wheel in the first row is skinny, and it also moves fast and undergoes, while the segmentation results of our MaskVOS are correct and robust. The target man in the second row is dancing in the street with various non-rigid movements, and our method could easily achieve accurate segmentation.

### E. Ablation Study

We analyze the effect of the individual components of our method and some configurations. Unless specified, we use the ResNet-50 as our backbone with 480p input resolution, pretrain on COCO, train with DAVIS17 and YouTube-VOS, and test on DAVIS17-val.

*1) Mask Utilization on Encoders:* We first ablate the mask impact on feature extractors, i.e., comparing all the instantiations (I1-I8). The matcher and decoder of them are the same as STM for reducing the extra influence and realizing

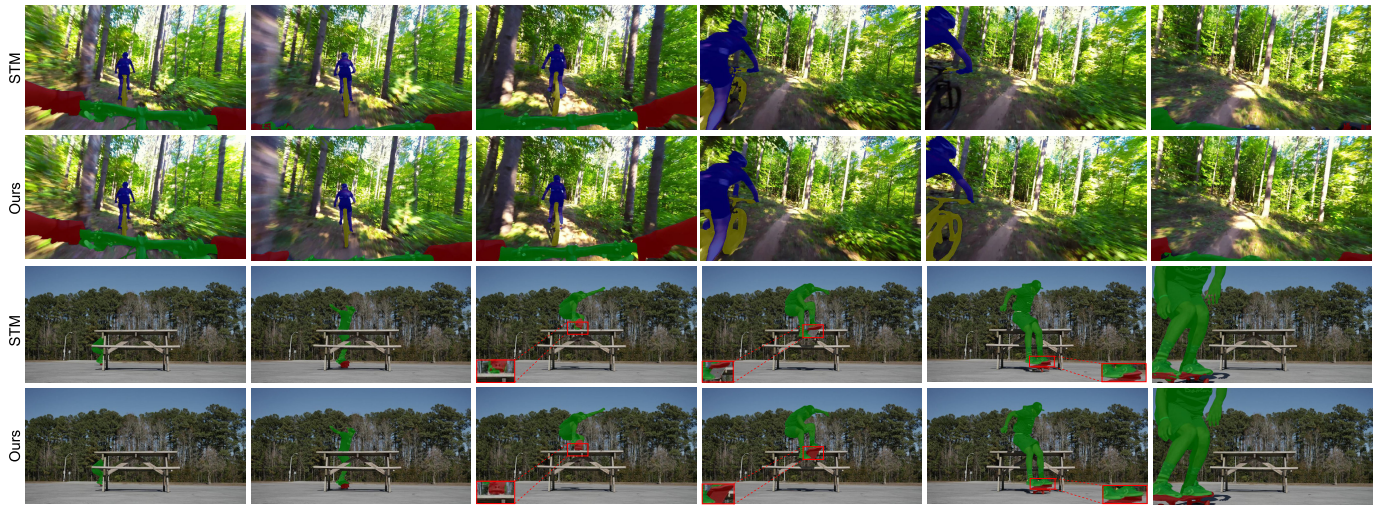


Fig. 5. Qualitative comparison of our method with the baseline STM [1] on the DAVIS2017. Some details are zoomed in with red rectangles. Our model handles some hard, challenging conditions like object occlusion, motion blur and dramatic appearance changes better than STM due to the full utilization of reference masks.

Fig. 6. Qualitative results of MaskVOS on the DAVIS17. The chosen scenes represent tough occasions, such as occlusion, shadows, non-rigid motion, small objects, complicated shapes and drastic changes in shape, which our method manages to handle.

fair comparison on the encoder part. We directly train these models on DAVIS17 and YouTube-VOS with input resolution $240 \times 427$ and test them on DAVIS17-val to fully explore the representation power of these encoders. The three parts of Table III show the results of early fusion, post fusion and multi-scale fusion, respectively. Comparing I2 *v.s.* (I3, I4) or I5 *v.s.* (I6, I8), we could find that extracting the features of masks brings a large performance improvement than using raw masks or simple downsampling. The reason is that mask representation is supervised and learned by the final VOS loss function for better results, while the raw masks do not. From I3 *v.s.* I4 and I6 *v.s.* I8, we observe that sharing the query and reference encoders but giving a stand-alone encoder for the mask is a superior configuration of the encoder, which reduces the frame encoder parameters and empowers better embedding for the mask. Besides, regarding the reference frame encoder (I8) rather than the mask encoder (I7) as the mainstream achieves better results since the mask encoder is usually a small network like ResNet-18, which has lower representation power than the frame encoder with a larger network. Moreover, the best multi-scale fusion of the reference mask and frame (I8) surpasses both the best early fusion (I1) and post fusion (I4), because the low-level, middle-level and high-level features are sufficiently interacted in multi-scale fusion. In conclusion, our I8 achieves the top performance with fewer parameters than most instantiations.

*2) Memory Bank Management:* We compare different memory management rules in Table IV. It could be found that

TABLE III

IMPACT OF MASK UTILIZATION ON ENCODERS. MODELS ARE TESTED WITH THE INPUT RESOLUTION OF 240P ON DAVIS17-VAL

| Models | Parameters (M) | $\mathcal{J}$ (%) | $\mathcal{F}$ (%) | $\mathcal{J}\&\mathcal{F}$ (%) |
|---|---|---|---|---|
| *I1* | 38.86 | 69.7 | 76.2 | 73.0 |
| *I2* | 38.61 | 67.5 | 72.8 | 70.1 |
| *I3* | 41.38 | 70.8 | 77.7 | 74.2 |
| *I4* | 41.36 | 70.3 | 77.0 | 73.6 |
| *I5* | 59.99 | 68.0 | 75.1 | 71.6 |
| *I6* | 44.75 | 70.0 | 76.9 | 73.5 |
| *I7* | 29.23 | 68.5 | 74.9 | 71.7 |
| ***I8*** | 36.24 | **71.3** | **77.8** | **74.6** |

TABLE IV

ABLATIONS ON MEMORY BANK MANAGEMENT. AFTER PRETRAINING ON COCO, MODELS ARE TRAINED ON DAVIS17 AND YOUTUBE-VOS, AND ARE TESTED WITH THE INPUT RESOLUTION OF 480P ON DAVIS17-VAL

| Reference Sets | $\mathcal{J}_\mathcal{M}$ (%) | $\mathcal{J}_\mathcal{R}$ (%) | $\mathcal{J}_\mathcal{D}$ (%) | $\mathcal{F}_\mathcal{M}$ (%) | $\mathcal{F}_\mathcal{R}$ (%) | $\mathcal{F}_\mathcal{D}$ (%) | $\mathcal{J}\&\mathcal{F}$ (%) |
|---|---|---|---|---|---|---|---|
| *First* | 66.8 | 75.2 | 27.3 | 73.9 | 82.1 | 30.9 | 70.4 |
| *Previous* | 70.6 | 80.8 | 18.5 | 76.5 | 87.6 | 21.6 | 73.6 |
| *First & Previous* | 77.9 | 88.5 | 11.6 | 84.4 | 94.0 | 16.3 | 81.2 |
| *Every five & First & Previous* | 80.0 | 90.5 | 7.1 | 86.7 | 95.8 | 11.6 | 83.3 |

TABLE V

EFFECTIVENESS OF THE PROPOSED MASK-ENHANCED MATCHER. AFTER THE PRETRAINING ON COCO, MODELS ARE TRAINED ON DAVIS17-TRAIN AND TESTED WITH THE INPUT RESOLUTION OF 480P ON DAVIS17-VAL

| Variants | $\mathcal{J}_\mathcal{M}$ (%) | $\mathcal{J}_\mathcal{R}$ (%) | $\mathcal{J}_\mathcal{D}$ (%) | $\mathcal{F}_\mathcal{M}$ (%) | $\mathcal{F}_\mathcal{R}$ (%) | $\mathcal{F}_\mathcal{D}$ (%) | $\mathcal{J}\&\mathcal{F}$ (%) |
|---|---|---|---|---|---|---|---|
| *w/o Q&R* | 77.1 | 89.2 | 12.9 | 82.6 | 92.8 | 18.4 | 79.9 |
| *w/ R* | 77.2 | 86.6 | 12.3 | 83.1 | 92.5 | 17.2 | 80.1 |
| *w/ Q&R* | 77.8 | 88.7 | 10.4 | 83.5 | 94.1 | 16.8 | 80.6 |

saving both the first and the previous frame into the memory is the most important, bringing significant performance improvements than using either one. This is because our model is strong enough to handle large appearance changes while being robust to drifting and error accumulation by effectively exploiting the memory. Besides, having every five frames saved in addition to the first and the previous frames further boosts performance with $+2.1\%$ gains. Therefore, we can save either the first and the previous frames in the memory for minimal memory consumption or add a new intermediate memory frame at every five frames for maximal accuracy. The latter is the configuration of our final model.

*3) Effectiveness of the Mask-Enhanced Matcher:* In Table V, we explore the impact of our proposed matcher, where models are first pretrained on COCO, then trained on DAVIS17-train and tested on DAVIS17-val. Specifically, we have three variants, (i) the ordinary matcher without using mask enhancement in both query and reference embeddings (w/o Q&R), (ii) just enhancing the reference embeddings (w/R), and (iii) enhancing both the query and reference embeddings (w/ Q&R). We observe that no mask-enhanced embeddings (i) decays accuracy by $-0.7\%$ over both aggregated embeddings (iii). Only enhancing reference (ii) brings $+0.2\%$ over (i) but is still worse than (iii) by $-0.5\%$. The results demonstrate that the mask enhancement is indeed beneficial in the matcher, and it is helpful in both query and reference embeddings. Our MaskVOS applies mask enhancement in both query and reference embeddings (iii) which has the best performance. We also visualize the attention maps and mask predictions of the proposed mask-enhanced matcher (iii) and the commonly-used matcher (i) in Fig. 1, from which we can see that the former is more robust for the distraction of background.

TABLE VI

IMPACTS OF DIFFERENT SIMILARITY COMPUTATIONS. AFTER PRETRAINING ON COCO, MODELS ARE TRAINED ON DAVIS17 AND YOUTUBE-VOS, AND ARE TESTED WITH THE INPUT RESOLUTION OF 480P ON DAVIS17-VAL

| Similarity types | $\mathcal{J}_\mathcal{M}$ (%) | $\mathcal{J}_\mathcal{R}$ (%) | $\mathcal{J}_\mathcal{D}$ (%) | $\mathcal{F}_\mathcal{M}$ (%) | $\mathcal{F}_\mathcal{R}$ (%) | $\mathcal{F}_\mathcal{D}$ (%) | $\mathcal{J}\&\mathcal{F}$ (%) |
|---|---|---|---|---|---|---|---|
| *L2* | 80.0 | 90.5 | 7.1 | 86.7 | 95.8 | 11.6 | 83.3 |
| *dot product* | 79.6 | 89.8 | 9.2 | 86.0 | 94.0 | 12.7 | 82.8 |

TABLE VII

ABLATIONS ON THE DIFFERENT BACKBONE. AFTER PRETRAINING ON COCO, MODELS ARE TRAINED ON DAVIS17 AND YOUTUBE-VOS, AND ARE TESTED WITH THE INPUT RESOLUTION OF 480P ON DAVIS17-VAL

| Backbone | $\mathcal{J}_\mathcal{M}$ (%) | $\mathcal{J}_\mathcal{R}$ (%) | $\mathcal{J}_\mathcal{D}$ (%) | $\mathcal{F}_\mathcal{M}$ (%) | $\mathcal{F}_\mathcal{R}$ (%) | $\mathcal{F}_\mathcal{D}$ (%) | $\mathcal{J}\&\mathcal{F}$ (%) | MACs (G) | Params (M) |
|---|---|---|---|---|---|---|---|---|---|
| *ResNet-50* | 80.0 | 90.5 | 7.1 | 86.7 | 95.8 | 11.6 | 83.3 | 192.1 | 36.4 |
| *Swin-small* | 82.0 | 91.7 | 5.4 | 89.0 | 96.2 | 9.2 | 85.5 | 266.6 | 59.8 |

TABLE VIII

ABLATIONS ON DIFFERENT INPUT RESOLUTIONS. AFTER PRETRAINING ON COCO, MODELS ARE TRAINED ON DAVIS17-TRAIN AND TESTED ON DAVIS17-VAL

| Input resolution | $\mathcal{J}_\mathcal{M}$ (%) | $\mathcal{J}_\mathcal{R}$ (%) | $\mathcal{J}_\mathcal{D}$ (%) | $\mathcal{F}_\mathcal{M}$ (%) | $\mathcal{F}_\mathcal{R}$ (%) | $\mathcal{F}_\mathcal{D}$ (%) | $\mathcal{J}\&\mathcal{F}$ (%) | MACs (G) | FPS |
|---|---|---|---|---|---|---|---|---|---|
| *240p* | 67.9 | 79.4 | 11.8 | 75.5 | 87.8 | 17.0 | 71.7 | 48.0 | 28.1 |
| *480p* | 77.8 | 88.7 | 10.4 | 83.5 | 94.1 | 16.8 | 80.6 | 192.1 | 11.1 |

*4) Similarity Calculation:* Table VI shows results for different ways of similarity calculation in the matcher. Two memory coverage types are considered here, L2 similarity and dot product. We observe that L2 similarity yields better performance by $+0.5\%$ than dot product, which is consistent with the conclusion of [3]. Therefore, we employ L2 similarity in our final model.

*5) Varying the Backbone:* Our MaskVOS has two backbone configurations, the traditional convolution network ResNet [51] and one recent typical transformer swin-transformer [52]. More specifically, ResNet-50 and swin-small are used for the frame encoder, and ResNet-18 and swin-tiny are used for the mask encoder. As shown in Table VII, the transformer-based model achieves better results with a large improvement of $+2.2\%$ over the convolution-based model. However, the transformer-based model will inevitably introduce more computational cost, as shown in Table VII. We provide the two backbone variants for different accuracy-efficiency requirements and adopt both configurations in MaskVOS for a fair comparison with previous methods.

*6) Varying the Input Resolutions:* We now ablate the input resolutions in Table VIII with 240p and 480p. The models are

first pretrained on COCO then trained on DAVIS17-train and tested on DAVIS17-val. The results are intuitive, i.e., a larger input size results in much better performance (+9.3%). On the other hand, increasing the input size will inevitably slow down the inference speed, from 28.1 FPS to 11.1 FPS. Therefore, we provide both two variants to adapt to different application requirements.

## V. Conclusion

This paper delves deeper into mask utilization in both the encoder and matcher of VOS. We first implement eight kinds of encoders in a unified platform to investigate effective strategies of reference masks and frames fusion in the encoder part. One of our designs yields the best results and indicates that (i) a stand-alone mask encoder is necessary for better mask representations, and (ii) multi-scale fusion is beneficial for sufficient interaction between the features of reference masks and frames. Moreover, we present a new mask-enhanced matcher to explore the mask impact on the matcher part, decreasing the background distraction and aggregating the locality of the matching process. A new network is then proposed, named MaskVOS, which consists of the mask-fused encoder, the mask-enhanced matcher and a standard decoder. Our simple yet effective network sufficiently exploits the reference masks and achieves state-of-the-art results on three commonly-used VOS datasets.

We hope this work will push a step further for the VOS research field, especially the previously neglected mask utilization problem. Besides, the unified codebase is also useful for this task since lots of models could be found and easily implemented in it. We believe the architectures and ideas discussed in this paper are successful in excavating the advantages of the reference masks. It could be extended for other video segmentation tasks such as video instance segmentation and video panoptic segmentation. We leave this as our future work.

## References


[1] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9226–9235.

[2] Y. Liang, X. Li, N. Jafari, and Q. Chen, "Video object segmentation with adaptive feature bank and uncertain-region refinement," 2020, *arXiv:2010.07958*.

[3] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1–14.

[4] Y. Li, N. Xu, J. Peng, J. See, and W. Lin, "Delving into the cyclic mechanism in semi-supervised video object segmentation," 2020, *arXiv:2010.12176*.

[5] W. Ge, X. Lu, and J. Shen, "Video object segmentation using global and instance embedding learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16836–16845.

[6] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2663–2672.

[7] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7376–7385.

[8] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg, "A generative appearance model for end-to-end video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8953–8962.

[9] H. Lin, X. Qi, and J. Jia, "AGSS-VOS: Attention guided single-shot video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3949–3957.

[10] Z. Wang, J. Xu, L. Liu, F. Zhu, and L. Shao, "RANet: Ranking attention network for fast video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3978–3987.

[11] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, "FEELVOS: Fast end-to-end embedding learning for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9481–9490.

[12] H. Wang, X. Jiang, H. Ren, Y. Hu, and S. Bai, "SwiftNet: Real-time video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1296–1305.

[13] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1–12.

[14] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.

[15] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 Davis challenge on video object segmentation," 2017, *arXiv:1704.00675*.

[16] N. Xu *et al.*, "YouTube-VOS: Sequence-to-sequence video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 585–601.

[17] P. Voigtlaender and B. Leibe, "Online adaptation of convolutional neural networks for video object segmentation," 2017, *arXiv:1706.09364*.

[18] K.-K. Maninis *et al.*, "Video object segmentation without temporal information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1515–1530, Jun. 2019.

[19] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele, "Lucid data dreaming for object tracking," in *Proc. DAVIS Challenge Video Object Segmentation-CVPR Workshops*, 2017, pp. 1–6.

[20] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 221–230.

[21] J. Luiten, P. Voigtlaender, and B. Leibe, "PReMVOS: Proposal-generation, refinement and merging for video object segmentation," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 565–580.

[22] A. Robinson, F. J. Lawin, M. Danelljan, F. S. Khan, and M. Felsberg, "Learning fast and robust target models for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7406–7415.

[23] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[24] Z. Zhang and H. Peng, "Deeper and wider Siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4591–4600.

[25] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6269–6277.

[26] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[27] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12549–12556.

[28] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6668–6677.

[29] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.

[30] X. Chen, Z. Li, Y. Yuan, G. Yu, J. Shen, and D. Qi, "State-aware tracker for real-time video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9384–9393.

[31] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6578–6588.

[32] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "VideoMatch: Matching based video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 54–70.

[33] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," 2020, *arXiv:2007.07020*.

[34] H. Xie, H. Yao, S. Zhou, S. Zhang, and W. Sun, "Efficient regional memory network for video object segmentation," 2021, *arXiv:2103.12934*.

[35] L. Hong, W. Zhang, L. Chen, W. Zhang, and J. Fan, "Adaptive selection of reference frames for video object segmentation," *IEEE Trans. Image Process.*, vol. 31, pp. 1057–1071, 2022.

[36] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by foreground-background integration," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 332–348.

[37] H. Seong, J. Hyun, and E. Kim, "Kernelized memory network for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 629–645.

[38] Y. Li, Z. Shen, and Y. Shan, "Fast video object segmentation using the global context module," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 735–750.

[39] L. Hu, P. Zhang, B. Zhang, P. Pan, Y. Xu, and R. Jin, "Learning position and target consistency for memory-based video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4144–4154.

[40] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "SSTVOS: Sparse spatiotemporal transformers for video object segmentation," 2021, *arXiv:2101.08833*.

[41] Y. Mao, N. Wang, W. Zhou, and H. Li, "Joint inductive and transductive learning for video object segmentation," 2021, *arXiv:2108.03679*.

[42] J. Mei, M. Wang, Y. Lin, Y. Yuan, and Y. Liu, "TransVOS: Video object segmentation with transformers," 2021, *arXiv:2106.00588*.

[43] K. Park, S. Woo, S. W. Oh, I. S. Kweon, and J.-Y. Lee, "Per-clip video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 1352–1361.

[44] J. Wu, Q. Liu, Y. Jiang, S. Bai, A. Yuille, and X. Bai, "In defense of online models for video instance segmentation," 2022, *arXiv:2207.10661*.

[45] B. Miao, M. Bennamoun, Y. Gao, and A. Mian, "Region aware video object segmentation with deep motion modeling," 2022, *arXiv:2207.10258*.

[46] G. Pei, F. Shen, Y. Yao, G.-S. Xie, Z. Tang, and J. Tang, "Hierarchical feature alignment network for unsupervised video object segmentation," 2022, *arXiv:2207.08485*.

[47] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5559–5568.

[48] M. Lan, J. Zhang, F. He, and L. Zhang, "Siamese network with interactive transformer for video object segmentation," 2021, *arXiv:2112.13983*.

[49] H. Chen, H. Wu, N. Zhao, S. Ren, and S. He, "Delving deep into many-to-many attention for few-shot video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14040–14049.

[50] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang, "Real-time fusion network for RGB-D semantic segmentation incorporating unexpected obstacle detection for road-driving images," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5558–5565, Oct. 2020.

[51] 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.

[52] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[53] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.


**Mengmeng Wang** received the B.S. and M.S. degrees in control science and engineering from Zhejiang University, Zhejiang, China, in 2015 and 2018, respectively, where she is currently pursuing the Ph.D. degree with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering. Her research interests include visual tracking, action recognition, computer vision, and deep learning.

**Jianbiao Mei** received the B.S. degree in control science and engineering from Zhejiang University, Zhejiang, China, in 2021, where he is currently pursuing the M.S. degree with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering. His research interests include video object segmentation, computer vision, and deep learning.

**Lina Liu** received the B.S. degree in automation from Zhejiang University in 2018, where she is currently pursuing the Ph.D. degree with the Department of Control Science and Engineering, Institute of Cyber-Systems and Control. Her research interests include computer vision and deep learning.

**Guanzhong Tian** (Member, IEEE) received the B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2010, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2021. He is currently a Research Associate with the Ningbo Research Institute, Zhejiang University. His research interests include computer vision, model compression, and embedded AI.

**Yong Liu** (Member, IEEE) received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Zhejiang, China, in 2001 and 2007, respectively. He is currently a Professor with the Institute of Cyber-Systems and Control, Zhejiang University. His main research interests include robot perception and vision, deep learning, big data analysis, multi-sensor fusion, machine learning, computer vision, information fusion, and robotics.

**Zaisheng Pan** received the B.S. and master's degrees in production process automation from Zhejiang University, Zhejiang, China, in 1993 and 1996, respectively. He is currently a Professor with the Institute of Cyber-Systems and Control, Zhejiang University. His main research interests include the Internet of Things, industrial big data, and industrial networks.