Learning Quality-aware Dynamic Memory for Video Object Segmentation

Yong Liu[1*], Ran Yu[1], Fei Yin[1], Xinyuan Zhao[2], Wei Zhao[2], Weihao Xia[3], and Yujiu Yang[1†]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University
[2] Huawei Technologies
[3] University College London
{liu-yong20,yu-r19}@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn

Abstract. Recently, several spatial-temporal memory-based methods have verified that storing intermediate frames and their masks as memory are helpful to segment target objects in videos. However, they mainly focus on better matching between the current frame and the memory frames without explicitly paying attention to the quality of the memory. Therefore, frames with poor segmentation masks are prone to be memorized, which leads to a segmentation mask error accumulation problem and further affect the segmentation performance. In addition, the linear increase of memory frames with the growth of frame number also limits the ability of the models to handle long videos. To this end, we propose a **Q**uality-aware **D**ynamic **M**emory **N**etwork (QDMN) to evaluate the segmentation quality of each frame, allowing the memory bank to selectively store accurately segmented frames to prevent the error accumulation problem. Then, we combine the segmentation quality with temporal consistency to dynamically update the memory bank to improve the practicability of the models. Without any bells and whistles, our QDMN achieves new state-of-the-art performance on both DAVIS and YouTube-VOS benchmarks. Moreover, extensive experiments demonstrate that the proposed Quality Assessment Module (QAM) can be applied to memory-based methods as generic plugins and significantly improves performance. Our source code is available at https://github.com/workforai/QDMN.

Keywords: video object segmentation, memory bank

1 Introduction

Given a video and the first frame's annotations of single or multiple objects, semi-supervised video object segmentation (Semi-VOS or One-shot VOS) aims at segmenting these objects in subsequent frames. Semi-VOS is one of the most challenging tasks in computer vision with many potential applications, including interactive video editing, augmented reality, and autonomous driving.

[*] This work was done during an internship at Huawei Technologies
[†] Corresponding author

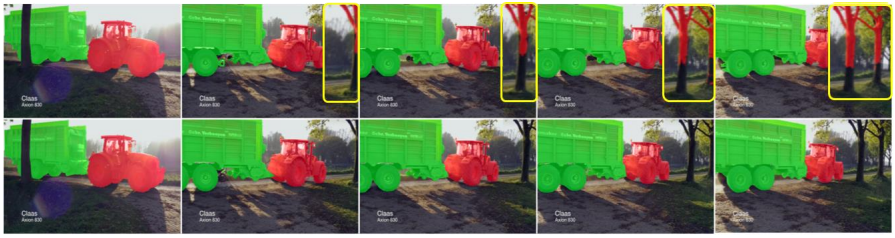

Fig. 1: Visual comparison of memory frames of different qualities. The first row shows the memory frames of MiVOS [6]. The second row shows the memory frames of our method. The yellow box area illustrates the error accumulation.

Unlike other segmentation tasks that aim to look for the relationship between features and specific categories, the critical problem of Semi-VOS lies in how to make full use of the spatial-temporal information to recognize the target objects. Consequently, the methods that perform matching with historical reference frames have received tremendous attention in recent years. Some works [40,50,52] utilize the first frame and the previous adjacent frame as references. Due to limited reference information, these approaches tend to fail miserably under challenging scenarios, *e.g.*, the target objects disappear for a while or are drastically deformed. To excavate more information, the Space-Time Memory Network (STM) [29] utilizes a memory network to memorize intermediate frames and their segmentation masks as references, which has been proved effective and has served as the current mainstream framework. Many approaches [35,21,6,14,46,36,42,7] further develop the feature extraction and memory readout process of STM and have achieved excellent performance.

However, these methods mainly focus on optimizing the matching process while ignoring the impact of the matching target, *i.e.,* memory bank, on the segmentation results. Specifically, previous methods select memory frames in a straightforward way, *i.e.*, storing at fixed frame intervals. This approach has two weaknesses: (1) Frames with poor segmentation results may be memorized and provide an erroneous reference for subsequent frames, which leads to an error accumulation problem. As shown in the first row of Fig. 1, if there are inaccurately segmented masks in the memory bank, the segmentation quality of subsequent frames will be greatly degraded. Such an observation inspires us to pay more attention to the design of the memory bank. Since the matching-based approaches rely on a memory bank to identify the target objects, the memory bank's quality (especially the correctness) is very important. (2) In existing methods, the size of the memory bank would infinitely expand with the growth of frame number, which makes the models incapable of handling long videos and greatly limits their practicality.

Therefore, the way of designing the memory bank is a significant issue for spatial-temporal memory-based methods. Generally speaking, we believe that the design of the memory bank should meet the following principles: (1) ***Accu-***

racy: *In a one-shot scenario, the memory bank should be composed of the annotated frame and frames that are segmented as accurately as possible to obtain correct supervision information.* (2) ***Temporal consistency:*** *Considering the continuity of motion, the state of objects in adjacent frames tends to be similar. In other words, the masks of adjacent frames are of great reference to the current frame.* Based on these two principles, we can selectively store frames with more reference information as memory and dynamically update the memory bank to handle videos of arbitrary length.

To this end, we propose a Quality-aware Dynamic Memory Network (QDMN), which introduces a simple but effective structure called Quality Assessment Module (QAM) in this paper to evaluate each frame's segmentation result and decide whether a frame can be added to the memory bank as a reference. Being aware of the segmentation quality limits the impact of noise and provides the accuracy credentials for dynamically updating the memory bank. Besides, since the objects in adjacent frames share a similar status to the current target, we introduce a temporal regularization to penalize the outdated memory. Extensive experiments demonstrate that the dynamic updating strategy of the memory bank designed according to the principles of accuracy and temporal consistency is reasonable and effective. By designing a high-quality memory bank and introducing temporal consistency, our method achieves new state-of-the-art performance on both DAVIS [33] and Youtube-VOS [47] benchmark without any bells and whistles. Furthermore, we also verify that memory-based methods can gain significant improvement by simply applying our QAM as a generic plugin for video object segmentation tasks.

Our contributions can be summarized as follows. Firstly, we pinpoint the design of the memory bank as the Achilles heel of the Semi-VOS task and propose the strategy for designing a high-quality memory bank. Secondly, we present QDMN for Semi-VOS, which can selectively memorize high-quality frames and take advantage of the temporal consistency. Thirdly, QDMN can effectively control the number of memory frames to avoid memory explosion. Experiments show that our method surpasses the existing methods on both DAVIS and YouTube-VOS datasets. Furthermore, QAM can be used as a generic plugin to improve memory-based methods.

2 Related Work

Propagation-based Methods. Propagation-based methods [39,9,8,49,1,15,41] treat semi-supervised video object segmentation as a mask propagation task. MaskTrack [31] concatenates the previous adjacent frame's segmentation mask with the current image as input and online fine-tunes the network. AGSS-VOS [24] proposes an attention-guided decoder to combine the instance-specific branch and instance-agnostic branch. Based on mask confidence and mask concentration, SAT [3] selectively propagates the entire image or local region to the next frame. The propagation-based method takes advantage of the strong prior provided by the previous adjacent frame. It can better deal with the appear-

ance change of the target object, but it has fatal shortcomings in the problem of occlusion and error accumulation.

Detection-based Methods. Detection-based methods divide the Semi-VOS task into three subtasks: detection, tracking and segmentation. DyeNet [20] utilizes RPN [34] to generate proposals and applies the re-identification module to perform matching. PReMVOS [26] uses Mask RCNN [12] to obtain coarse masks and performs optical flow, re-identification to achieve good performance. Huang *et al* [16] and Sun *et al* [38] integrate segmentation into tracking with a dynamic template bank. Detection-based methods rely heavily on the detectors, which dramatically limits the performance of such methods.

Matching-based Methods. Matching-based methods perform matching between reference frames and the current frame to identify target objects, which has raised great attention for excellent performance and robustness. PML [4] proposes a pixel-level embedding network with the nearest neighbor classifier. FEELVOS [40] and CFBI [50] perform global and local matching with the first frame and the previous adjacent frame, respectively. AOT [51] associates multiple target objects into the same embedding space by employing an identification mechanism. STM [29] leverages the memory network to memorize intermediate frames as references, which has been proved effective and has served as the current mainstream framework. Based on STM, KMN [35] and RMNet [46] propose to perform local-to-local matching instead of non-local. SwiftNet [42] and AFB-URR [23] reduce memory duplication redundancy by calculating the similarity between query and memory. LCM [14] emphasizes the importance of the first frame and the previous adjacent frame. STCN [7] improves the feature extraction and performs reasonable matching by decoupling the image and masks. Following the memory-based idea, there are still many variants of STM, such as JOINT [27], EGMN [25], MiVOS [6], DMN-AOA [22], HMMN [36], and so on.

Although these methods have achieved great performance, they mainly focus on better matching the current frame with the memory frames. In other words, previous works dedicate to optimizing the matching process while neglecting the importance of matching with the correct object. Besides, they do not take into account that the size of the memory bank grows linearly with the length of the video, which greatly impacts the application of the models in real scenarios due to the hardware memory limitation.

3 Method

3.1 Overview

The overall architecture of our QDMN is shown in Fig. 2. Similar to STM [29], during video processing, the current frame (t-th frame) is considered as the query, and the past reference frames with segmentation masks are considered as the memory. The query and memory are encoded into pairs of key and value

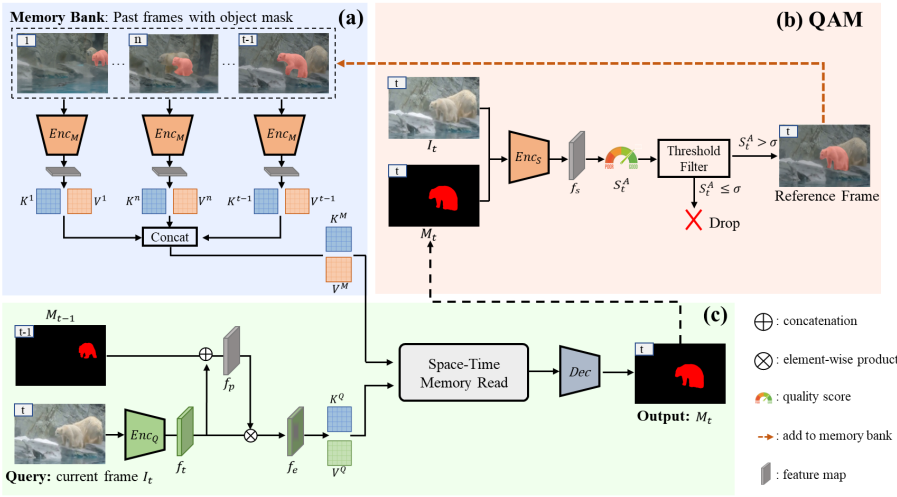

Fig. 2: Overview of QDMN. (a) is the feature extraction of the reference frames in the memory bank. (b) QAM is the module used to evaluate whether the current frame can be added to the memory bank. (c) is the pipeline for predicting the segmentation result of the current frame I_t .

maps through visual encoders and corresponding convolution layers. To highlight the temporal consistency of video, the query feature f_t is first enhanced with the prior mask to obtain the enhanced feature f_e. Then the enhanced feature is encoded into pairs of key K^Q and value V^Q through corresponding convolution layers. The Space-Time Memory Read block performs pixel-level matching between K^Q and the memory key K^M. The relative matching similarity is used to address the memory value V^M, and the corresponding values are combined to the decoder for segmentation. Finally, the Quality Assessment Module (QAM) evaluates the quality of the segmentation result and decides whether the query frame can become a memory frame.

3.2 Quality Assessment Module

Designing the memory bank is a significant issue for memory network-based methods. For existing strategy, frames with erroneous masks may be memorized, which leads to an error accumulation problem. To alleviate this problem and ensure the accuracy of the memory bank, inspired by [17,18], we propose the Quality Assessment Module (QAM) to evaluate the segmentation quality and decide whether a frame can be added to the memory bank as a reference.

QAM is a simple structure but effective module composed of a score encoder, four convolution layers, and two MLP layers. It takes the query image $\mathbf{I}_t$ and its segmentation mask $\mathbf{M}_t$ as input and outputs the predicted quality scores. Since the feature extraction process of the score encoder Enc_s is the same as that

Fig. 3: Illustrations of segmentation masks with different quality scores. The three rows represent the ground truth, segmentation results, and the quality scores predicted by QAM, respectively.

of the memory encoder Enc_M (both takes images with segmentation masks as input), we directly use the memory encoder as the score encoder, which helps to save calculations and parameters. Specifically, the structure of the score encoder Enc_s and the memory encoder Enc_M is the same, and the parameters are shared. The QAM first takes the query image $\mathbf{I}_t \in \mathbb{R}^{3\times H\times W}$ and its segmentation mask $\mathbf{M}_t \in \mathbb{R}^{1\times H\times W}$ into the score encoder to obtain the score feature map $f_s \in \mathbb{R}^{C\times H/16\times W/16}$, where $H\times W$ are resolutions of the input image. Then, f_s is input to the convolution layers and fully connected layers to learn the segmentation quality score $\mathbf{S}_t^A$ for the current frame. The process of segmentation quality assessment can be expressed as:

$$f_s = Enc_s(\mathbf{I}_t \oplus \mathbf{M}_t); \quad \mathbf{S}_t^A = Fc(Conv(f_s)), \tag{1}$$

where $\oplus$ denotes the concatenation operation. t is the index of the current frame. $Conv$ and Fc denote convolution and fully connected layers with sigmoid non-linear function, respectively.

During training, the target value of the quality score is defined as mask IoU between the segmentation mask and ground truth. The specific calculation process is as follows:

$$loss = \frac{1}{N}\sum_{i=1}^{N}(S_i^A - maskIoU(M_i, GT_i))^2, \tag{2}$$

where S_i^A represents the quality score of the segmentation result for i-th object, M_i indicates the segmentation result, GT_i is the ground truth. N indicates the total number of objects.

Since QAM evaluates the segmentation quality for each object individually, we take the average of all object scores in one frame as the quality score of this frame. In addition, considering that the segmentation difficulty varies for different video scenes, we normalize the quality scores of all frames in a video to better measure the relative quality of the segmentation results, which helps to memorize more helpful information under challenging scenarios. Specifically,

the final quality score of each frame is its initial predicted score divided by the score of the first frame. Formally, the process can be expressed as:

$$\bar{\mathbf{S}}_t^A = \frac{\frac{1}{N}\sum_{i=1}^{N}\mathbf{S}_{t_i}^A}{\bar{\mathbf{S}}_1^A}, \tag{3}$$

where N represents the total number of objects in the t-th frame, $\bar{S}_t^A$ indicates the quality score of the segmentation result in frame t, $\bar{S}_1^A$ represents the quality score of the first frame.

Figure 3 shows some visualization results of the quality assessment, the first two columns are the same video, and the last two columns represent another video. We can observe that the driver is considered part of the car in the first column, which is a bad case. The pink zebra in the third column is not recognized, and the orange zebra is matched with similar background objects.

For the hard case, our QAM identifies these suboptimal results well, which shows that the segmentation accuracy of a frame is consistent with its quality score. Extensive experiments also verify this. With QAM, the memory bank can selectively memorize frames whose quality scores are higher than the memory threshold σ, that is, frames with accurate segmentation masks. In this way, even if a frame is poorly segmented owing to fast object motion or other factors, it will not affect the subsequent frames or cause error accumulation.

3.3 Dynamically Updated Memory Bank

The infinite increase of the memory frames with the growth of frame number greatly limits the practicability of the model in the real-world scenario. Thus, it is necessary to limit the size of the memory bank and update it dynamically to adapt to new scenarios.

Due to the temporal consistency of video, the appearance of the target objects in adjacent frames is similar. The masks of adjacent frames are more instructive for the segmentation of current frame. Combining the above analysis and considering accuracy, we suggest dynamically updating the memory bank in accordance with these two principles (Algorithm.1). Specifically, when the memory bank reaches a certain storage limit, we will dynamically update the memory bank to handle different video scenes. For quantifying the temporal consistency and measuring the distance between each memory frame and the current frame, we compute the temporal consistency score $\mathbf{S}^C$ as:

$$\mathbf{S}_k^C = e^{-|t-k|}, \tag{4}$$

Algorithm: 1 Pseudocode of Dynamic Memory Bank

Input : memory bank $Memory$, video frames sequence $\{I_t\}$ of length L

$t = 2$ # the ground truth mask of the first frame is given
$j = 1$ # the relative index of memory frames
while $t \leq L$ **do**
  if $S_t^A \geq \sigma$ **then**
    # to filter the inaccurately segmented frames
    $j = j + 1$
    if $len(Memory) \leq \beta$ **then**
      $Memory.add(\{j : [I_t, M_t, S_t^A]\})$ # store I_t, M_t, S_t^A to the j position in memory.
    else
      $S_{min}^R, id_{min} = inf, inf$
      for k in $Memory.keys()$ **do** # k is the relative index of the frame in the memory bank
        $S_k^C = exp(k - j)$
        $S_k^R = S_k^A + S_k^C$
        if $S_k^R < S_{min}^R$ **then**
          $S_{min}^R = S_k^R$, $id_{min} = k$
        end if
      end for
      $Memory.del(id_{min})$ # remove the memory frame with the lowest reference score S^R
      $Memory.add(\{j : [I_t, M_t, S_t^A]\})$
    end if
  end if
  $t = t + 1$
end while

where k is the index of each memory frame, t is the index of the current frame.

Based on the accuracy score $\mathbf{S}^A$ and the temporal consistency score $\mathbf{S}^C$, the reference score of each memory frame in the memory bank can be calculated by $\mathbf{S}_k^R = \tilde{\mathbf{S}}_k^A + \mathbf{S}_k^C$. By removing the memory frames with the lowest reference score, the memory bank is dynamically updated to handle different video scenarios and prevent the memory explosion problem.

3.4 Prior Enhancement Strategy

In addition to considering temporal consistency when designing a memory bank, we further utilize the prior provided by the previous adjacent frame to enhance temporal information. We adopt a similar module structure to SCM [54] to introduce the prior information from the previous adjacent frame. Instead of introducing spatial constraint in the decoder like SCM, we utilize the prior information in the query encoding process to better learn the target object's appearance feature and avoid over-reliance on the prior information.

Specifically, in the query encoding process, the segmentation mask of the previous adjacent frame $\mathbf{M}_{t-1} \in \mathbb{R}^{1\times H\times W}$ is downsampled and concatenated with the query's embedding $f_t \in \mathbb{R}^{C\times H/16\times W/16}$. Then the resultant feature goes through convolution and non-linear function to fuse information between channels, through which a prior feature map $f_p \in \mathbb{R}^{1\times H/16\times W/16}$ is produced. Finally, we perform an element-wise product between f_p and f_t to get the enhanced feature $f_e \in \mathbb{R}^{C\times H/16\times W/16}$. Formally, the process can be expressed as the following equation:

$$f_e = Conv(f_t \oplus \mathbf{M}_{t-1}) \otimes f_t. \tag{5}$$

Furthermore, we find that it is better to provide weak prior (mentioned above) than strong prior (masks of the previous frame have a great influence on the feature of the current frame). We found two primary reasons through experiments: the first one is that the prior information may be noisy, and providing a strong prior may lead to error accumulation; the second one is that providing strong prior makes the model overly dependent on it, which weakens its ability to extract features and identify objects. Table 6 shows the disadvantages of providing strong prior under challenging scenarios. In Section 5.3, we will describe the specific approach of providing strong prior.

3.5 Memory Read and Decoder

In the Space-Time Memory Read block [29], soft weights are first computed by measuring the similarities between query key K^Q and memory key K^M. Then the memory value V^M is retrieved by a weighted summation with the soft weights and concatenated with query value V^Q to get the output y. This operation can be summarized as:

$$y_i = V_i^Q \oplus \frac{1}{Z}\sum_{\forall j} \mathcal{D}(K_i^Q, K_j^M)V_j^M, \tag{6}$$

where i and j are the index of the query and the memory location, $Z = \sum_{\forall j} \mathcal{D}(K_i^Q, K_j^M)$ is the normalizing factor. $\mathcal{D}$ denotes the similarity measure (in our experiment is dot product).

Our decoder stays close to that of [55,29]. The decoder takes the output y of the Space-Time Memory Read block as input and predicts the object masks. It consists of an ASPP layer [2], a residual block, and two upsample blocks that upscale the feature map to the initial image size.

4 Implementation Details

Following the training strategy in MiVOS [6], we first pretrain our model on static image datasets [43,37,53,5,19] and then perform main training on YouTube-VOS and DAVIS datasets. Besides, we also experiment with the synthetic dataset BL30K proposed in MiVOS, which is not used unless otherwise specified. During pretraining, each image is expanded into a pseudo video of three frames by random affine, horizontal flip, color and brightness augmentation. We randomly pick three frames in chronological order (with a ground-truth mask for the first frame) from a video to form a training sample in the main training. The range of random sampling varies with the training process. In the intermediate period of training, the sampling range is set larger to improve the robustness of the model, while at the end of the training, it is set smaller to narrow the gap between training and inference. Our models are trained end-to-end with two 32GB Tesla V100 GPUs with the Adam optimizer in PyTorch. The batch size is set to 28 during pretraining and 16 during main training. We adopt ResNet-50 [13] as backbone for all encoders. Bootstrapped cross-entropy loss [6] is used for segmentation, and MSE loss is used for quality score evaluation. The initial learning rate is 2e-5. During inference, we choose the memory threshold σ of 0.8 by default. Ablation studies are conducted on a single 1080Ti GPU and DAVIS 2017 validation set in default.

5 Experiments

5.1 Comparisons with State-of-the-Art Methods

DAVIS 2016 [32] is a single object benchmark for video object segmentation. As shown in Table 1, QDMN trained without synthetic dataset still outperforms most previous methods (**91.0** $\mathcal{J}\&\mathcal{F}$). With synthetic training data, QDMN surpasses all existing methods and achieves the performance of **92.0** $\mathcal{J}\&\mathcal{F}$.
DAVIS 2017 [33] is a multiple objects extension of DAVIS 2016. In the Table 1, QDMN achieves an average score of **84.6** and **85.6** for training without synthetic data and with synthetic data, respectively. What's more, we also test our model on the challenging DAVIS 2017 testing split set. It achieves the best performance (**81.9**) compared to all previous methods.
YouTube-VOS [47] is a large-scale benchmark for video object segmentation. As shown in Table 2, without synthetic training data, our QDMN also achieves

Table 1: Comparison with other methods on DAVIS dataset. '*' indicates using synthetic training dataset [6].

Method	DAVIS2016			DAVIS2017 val			DAVIS2017 test-dev		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
RANet [44]	86.6	87.6	87.1	63.2	68.2	65.7	53.4	56.2	55.3
FEELVOS [40]	81.1	82.2	81.7	69.1	74.0	71.5	55.2	60.5	57.8
RGMP [28]	81.5	82.0	81.8	64.8	68.6	66.7	51.3	54.4	52.8
DMVOS [45]	88.0	87.5	87.8	-	-	-	-	-	-
STM [29]	88.7	89.9	89.3	79.2	84.3	81.8	69.3	75.2	72.2
KMN [35]	89.5	91.5	90.5	80.0	85.6	82.8	74.1	80.3	77.2
CFBI [50]	88.3	90.5	89.4	79.1	84.6	81.9	71.1	78.5	74.8
GIEL [11]	-	-	-	80.2	85.3	82.7	72.0	78.3	75.2
SwiftNet [42]	90.5	90.3	90.4	78.3	83.9	81.1	-	-	-
RMNet [46]	88.9	88.7	88.8	81.0	86.0	83.5	71.9	78.1	75.0
SSTVOS [10]	-	-	-	79.9	85.1	82.5	-	-	-
LCM [14]	89.9	91.4	90.7	80.5	86.5	83.5	74.4	81.8	78.1
MiVOS [6]	87.8	90.0	88.9	80.5	85.8	83.1	72.6	79.3	76.0
MiVOS* [6]	89.7	92.4	91.0	81.7	87.4	84.5	74.9	82.2	78.6
JOINT [27]	-	-	-	80.8	86.2	83.5	-	-	-
RPCMVOS [48]	87.1	94.0	90.6	81.3	86.0	83.7	75.8	82.6	79.2
DMN-AOA [22]	-	-	-	81.0	87.0	84.0	74.8	81.7	78.3
HMMN [36]	89.6	92.0	90.8	81.9	87.5	84.7	74.7	82.5	78.6
STCN [7]	90.8	92.5	91.6	82.2	88.6	85.4	72.7	79.6	76.1
AOT-L [51]	89.7	92.3	91.0	80.3	85.7	83.0	75.3	82.3	78.8
QDMN (Ours)	**90.2**	**91.7**	**91.0**	**81.8**	**87.3**	**84.6**	**74.2**	**81.2**	**77.7**
QDMN* (Ours)	**90.7**	**93.2**	**92.0**	**82.5**	**88.6**	**85.6**	**78.1**	**85.4**	**81.9**

state-of-the-art performance (**83.0**). If we use synthetic data for training, the overall score of QDMN will be boosted to **83.8**.

Qualitative results. The qualitative comparison between baseline and our QDMN are shown in Fig. 4. We show the performance on two challenging scenarios, *i.e.*, occlusion scenes and similar objects. Both STM [29] and MiVOS [6] have lost targets in the occlusion scene. STM lost targets in the scene with similar objects, while MiVOS identified other objects incorrectly. In contrast, our method can achieve satisfactory performance in challenging scenarios.

5.2 Generic Plugins

To further prove the effectiveness of our proposed QAM, we apply it as a general plugin to other methods. The results on the DAVIS2017 validation set are shown in Table 3 (the baseline performance is our re-implementation results). It can be seen that with QAM, the performance of these methods has been significantly boosted. Besides, QAM is easy to be deployed on other methods, and we hope that the QAM would shed light on the studies of related fields that need to memorize reference information.

Table 2: Evaluation on YouTube-VOS 2018 val set. Seen and Unseen denote whether the categories exist in the training set. $\mathcal{G}$ is averaged overall score.

Methods	Seen		Unseen		$\mathcal{G}$
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	
STM [29]	79.7	84.2	72.8	80.9	79.4
AFB-URR [23]	78.8	83.1	74.1	82.6	79.6
GCM [21]	72.6	75.6	68.9	75.7	73.2
KMN [35]	81.4	85.6	75.3	83.3	81.4
G-FRTM [30]	68.6	71.3	58.4	64.5	65.7
SwiftNet [42]	77.8	81.8	72.3	79.5	77.8
GIEL [11]	80.7	85.0	75.0	81.9	80.6
SSTVOS [10]	80.9	-	76.6	-	81.8
RMNet [46]	82.1	85.7	75.7	82.4	81.5
LCM [14]	82.2	86.7	75.7	83.4	82.0
MiVOS [6]	80.0	84.6	74.8	82.4	80.4
MiVOS* [6]	81.1	85.6	77.7	86.2	82.6
JOINT [27]	81.5	85.9	78.7	86.5	83.1
HMMN [36]	82.1	87.0	76.8	84.6	82.6
DMN-AOA [22]	82.5	86.9	76.2	84.2	82.5
STCN [7]	81.9	86.5	77.9	85.7	83.0
AOT-L [51]	82.5	87.5	77.9	86.7	83.7
QDMN (Ours)	**82.0**	**86.8**	**77.5**	**85.5**	**83.0**
QDMN* (Ours)	**82.7**	**87.5**	**78.4**	**86.4**	**83.8**

5.3 Ablation Study

The effectiveness of QAM. To demonstrate the effectiveness of the QAM, we conduct specific analyses from three dimensions.

(1) Accuracy of the predicted scores. We perform a histogram visualization of the distribution of the ground truth mask IoU and prediction scores at 0.05 intervals(Fig. 5). When multiple objects are in a frame, the average is taken. We can see that the quality score and ground truth mask IoU are positively correlated, which verifies the accuracy of the scores predicted by QAM.

(2) Memory Threshold. We test different memory thresholds σ on DAVIS 2017 test-dev set, and the results are shown in Fig. 6. We can see that it will hurt the segmentation effect if the threshold is set too high or too low. The reason is that if the threshold σ is too high, only a few intermediate frames will be memorized, leading to losing a lot of helpful information; if the σ is too low, the model may memorize some incorrect noise information. Besides, the performance is worst when the memory threshold is 0 (at this time, QAM does not filter poor segmentation masks), which proves the motivation of the QAM is correct.

(3) Applying QAM only at inference stage. To further prove that filtering out inaccurately segmented frames has a beneficial effect on segmentation, we construct experiments that adding QAM only at the inference stage. Specifically,

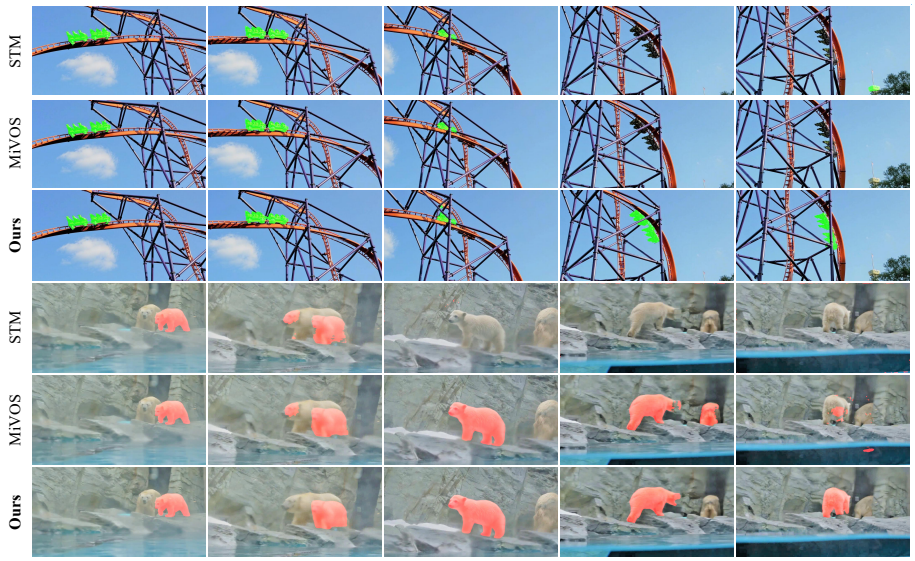

Fig. 4: Visual comparison of QDMN with baseline methods.Each row demonstrates five frames sampled from a video sequence.

for QAM, we load its parameters trained in QDMN. For other parts, we load the weights of the initial model (trained without QAM). As shown in Table 4, the performance of all vanilla models has been improved after adding QAM, which shows the importance of filtering poorly segmented frames.

Table 4: The effect of adding QAM only in *the inference stage*

Methods	$\mathcal{J}\&\mathcal{F}_{(\sigma=0)}$	$\mathcal{J}\&\mathcal{F}_{(\sigma=0.8)}$
STM	81.5	**82.5**↑
KMN	82.6	**83.4**↑
MiVOS	82.7	**83.5**↑

Table 5: Ablation study of proposed components.

QAM	PEM	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
		80.3	85.5	82.9
✓		81.7	87.1	**84.3**↑
	✓	81.1	86.1	**83.6**↑
✓	✓	81.8	87.3	**84.6**↑

Component Analysis. We analyze the effectiveness of our modules in Table 5. PE represents the prior enhancement strategy introduced to highlight temporal consistency. As shown in the table, both the QAM and PE bring remarkable performance improvement.

Dynamic Memory Updating Strategy. Due to the lack of a widely used large-scale long video dataset in this field, we choose to demonstrate the effectiveness of our proposed memory bank dynamic updating strategy by compress-

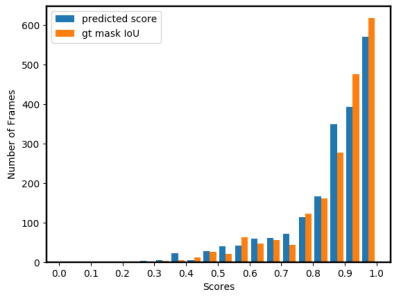

Fig. 5: Distribution of the prediction score and the ground truth mask IoU.

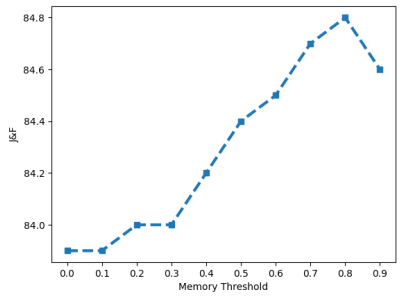

Fig. 6: The quantitative results of different memory threshold σ.

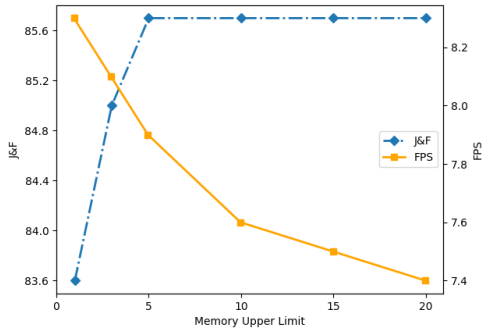

Fig. 7: The performance for different memory upper limit.

Methods	w / QAM	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
STM [29]		78.8	84.2	81.5
	✓	81.0	86.2	**83.6**$^{\uparrow}$
KMN [35]		79.7	85.5	82.6
	✓	81.9	87.4	**84.7**$^{\uparrow}$
STCN [7]		81.5	87.7	84.6
	✓	82.5	88.7	**85.6**$^{\uparrow}$

Table 3: Applying QAM as general plugin. w / QAM indicates that whether the QAM is deployed on this method.

ing the upper limit of the memory. As shown in Figure 7, The segmentation effect remains unaffected even at low memory upper limit, and the speed is improved as a result of our memory bank design strategy. The similar phenomenon is observed on the YouTube-VOS set, which illustrates the effectiveness of our dynamic updating strategy.

Besides, we also perform analysis on long videos (without annotations). We find that previous memory network methods store up to about 70 frames and the memory explosion occurs, which greatly limits the practicability. But QDMN can handle videos of arbitrary length by setting upper memory limit and dynamically updating the memory. What's more, the FPS of previous methods will drop from 14 to about 2 before memory explodes, while the FPS of QDMN will stay around 7 after the initial drop (assuming the upper memory limit is 25).

Enhancement Strategy. For PE, we directly concatenate the prior mask with the deepest layer feature of the current frame to provide a weak prior. In contrast, we also try to provide a strong prior. Specifically, we extract the feature of the prior mask and fuse it with the middle layer features of the current frame. After convolution and downsampling, the fused features are added to the deepest layer features of the current frame. Compared with the current enhancement strategy, this approach can significantly enhance the influence of the prior mask. However, although this approach works well in common scenarios, the performance drops significantly under challenging situations, as shown in Table 6. The reason for this phenomenon is that the strong prior makes the model overly dependent on it, which weakens the model's ability to recognize objects.

Table 6: Ablation study of different enhancement strategy. "Weak" means providing weak prior (PE). "Strong" means providing strong location prior.

Strategy	DAVIS			YouTube-VOS		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{G}$
Weak	81.8	87.3	84.6	79.8	86.2	**83.0**
Strong	82.4	87.9	**85.2**	77.5	83.8	80.7

Speed Analysis. We also experiment with the impact of the proposed modules on the inference speed. With our modules, the FPS of baseline has changed from **8.6** to **7.8** on DAVIS2017 val set. The increased running time brought by QAM and PE is nearly negligible (no more than 10%), mainly because we directly use the feature extracted by the memory encoder for quality assessment.

6 Conclusion

In this paper, we propose that the design of the memory bank should follow the principles of accuracy and temporal consistency. To support this, we introduce a Quality-aware Dynamic Memory Network (QDMN) for semi-supervised video object segmentation, which selectively memorizes accurately segmented intermediate frames as references and emphasizes video temporal consistency. Without bells and whistles, our QDMN achieves new state-of-the-art performance on the popular benchmark YouTube-VOS and DAVIS with almost no additional inference time. Furthermore, the QAM also has a remarkable improvement for other approaches as a general plugin.

Acknowledgments.

This research was supported in part by the National Natural Science Foundation of China under Grant No. U1903213, the Shenzhen Key Laboratory of Marine IntelliSense and Computation (NO. ZDSYS20200811142605016.)

References

1. Caelles, S., Maninis, K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Gool, L.V.: One-shot video object segmentation. In: CVPR. pp. 5320–5329 (2017) 3
2. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI pp. 834–848 (2018) 9
3. Chen, X., Li, Z., Yuan, Y., Yu, G., Shen, J., Qi, D.: State-aware tracker for real-time video object segmentation. In: CVPR. pp. 9381–9390 (2020) 3
4. Chen, Y., Pont-Tuset, J., Montes, A., Gool, L.V.: Blazingly fast video object segmentation with pixel-wise metric learning. In: CVPR. pp. 1189–1198 (2018) 4
5. Cheng, H.K., Chung, J., Tai, Y., Tang, C.: Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In: CVPR. pp. 8887–8896 (2020) 9
6. Cheng, H.K., Tai, Y., Tang, C.: Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. arXiv preprint arXiv:2103.07941 (2021) 2, 4, 9, 10, 11
7. Cheng, H.K., Tai, Y., Tang, C.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. arXiv preprint arXiv:2106.05210 (2021) 2, 4, 10, 11, 13
8. Cheng, J., Tsai, Y., Hung, W., Wang, S., Yang, M.: Fast and accurate online video object segmentation via tracking parts. In: CVPR. pp. 7415–7424 (2018) 3
9. Cheng, J., Tsai, Y., Wang, S., Yang, M.: Segflow: Joint learning for video object segmentation and optical flow. In: ICCV. pp. 686–695 (2017) 3
10. Duke, B., Ahmed, A., Wolf, C., Aarabi, P., Taylor, G.W.: Sstvos: Sparse spatiotemporal transformers for video object segmentation. In: CVPR. pp. 5912–5921 (2021) 10, 11
11. Ge, W., Lu, X., Shen, J.: Video object segmentation using global and instance embedding learning. In: CVPR. pp. 16836–16845 (2021) 10, 11
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV. pp. 2980–2988 (2017) 4
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 9
14. Hu, L., Zhang, P., Zhang, B., Pan, P., Xu, Y., Jin, R.: Learning position and target consistency for memory-based video object segmentation. arXiv preprint arXiv:2104.04329 (2021) 2, 4, 10, 11
15. Hu, Y., Huang, J., Schwing, A.G.: Maskrnn: Instance level video object segmentation. In: NIPS. pp. 325–334 (2017) 3
16. Huang, X., Xu, J., Tai, Y., Tang, C.: Fast video object segmentation with temporal aggregation network and dynamic template matching. In: CVPR. pp. 8876–8886 (2020) 4
17. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring R-CNN. In: CVPR. pp. 6409–6418 (2019) 5
18. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y.: Acquisition of localization confidence for accurate object detection. In: ECCV. pp. 816–832 (2018) 5
19. Li, X., Wei, T., Chen, Y.P., Tai, Y., Tang, C.: FSS-1000: A 1000-class dataset for few-shot segmentation. In: CVPR. pp. 2866–2875 (2020) 9
20. Li, X., Loy, C.C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: ECCV. pp. 93–110 (2018) 4

21. Li, Y., Shen, Z., Shan, Y.: Fast video object segmentation using the global context module. In: ECCV. pp. 735–750 (2020) 2, 11
22. Liang, S., Shen, X., Huang, J., Hua, X.S.: Video object segmentation with dynamic memory networks and adaptive object alignment. In: ICCV. pp. 8065–8074 (2021) 4, 10, 11
23. Liang, Y., Li, X., Jafari, N.H., Chen, J.: Video object segmentation with adaptive feature bank and uncertain-region refinement. In: NIPS (2020) 4, 11
24. Lin, H., Qi, X., Jia, J.: AGSS-VOS: attention guided single-shot video object segmentation. In: ICCV. pp. 3948–3956 (2019) 3
25. Lu, X., Wang, W., Danelljan, M., Zhou, T., Shen, J., Gool, L.V.: Video object segmentation with episodic graph memory networks. In: ECCV. pp. 661–679 (2020) 4
26. Luiten, J., Voigtlaender, P., Leibe, B.: Premvos: Proposal-generation, refinement and merging for video object segmentation. In: ACCV. pp. 565–580 (2018) 4
27. Mao, Y., Wang, N., Zhou, W., Li, H.: Joint inductive and transductive learning for video object segmentation. arXiv preprint arXiv:2108.03679 (2021) 4, 10, 11
28. Oh, S.W., Lee, J., Sunkavalli, K., Kim, S.J.: Fast video object segmentation by reference-guided mask propagation. In: CVPR. pp. 7376–7385 (2018) 10
29. Oh, S.W., Lee, J., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV. pp. 9225–9234 (2019) 2, 4, 8, 9, 10, 11, 13
30. Park, H., Yoo, J., Jeong, S., Venkatesh, G., Kwak, N.: Learning dynamic network using a reuse gate function in semi-supervised video object segmentation. In: CVPR. pp. 8405–8414 (2021) 11
31. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR. pp. 3491–3500 (2017) 3
32. Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L.V., Gross, M.H., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR. pp. 724–732 (2016) 9
33. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbelaez, P., Sorkine-Hornung, A., Gool, L.V.: The 2017 DAVIS challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017) 3, 9
34. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015) 4
35. Seong, H., Hyun, J., Kim, E.: Kernelized memory network for video object segmentation. In: ECCV. pp. 629–645 (2020) 2, 4, 10, 11, 13
36. Seong, H., Oh, S.W., Lee, J., Lee, S., Lee, S., Kim, E.: Hierarchical memory matching network for video object segmentation. arXiv preprint arXiv:2109.11404 (2021) 2, 4, 10, 11
37. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended CSSD. TPAMI pp. 717–729 (2016) 9
38. Sun, M., Xiao, J., Lim, E.G., Zhang, B., Zhao, Y.: Fast template matching and update for video object tracking and segmentation. In: CVPR. pp. 10788–10796 (2020) 4
39. Tsai, Y., Yang, M., Black, M.J.: Video segmentation via object flow. In: CVPR. pp. 3899–3908 (2016) 3
40. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.: FEELVOS: fast end-to-end embedding learning for video object segmentation. In: CVPR. pp. 9481–9490 (2019) 2, 4, 10
41. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017) 3

42. Wang, H., Jiang, X., Ren, H., Hu, Y., Bai, S.: Swiftnet: Real-time video object segmentation. In: CVPR. pp. 1296–1305 (2021) 2, 4, 10, 11
43. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: CVPR. pp. 3796–3805 (2017) 9
44. Wang, Z., Xu, J., Liu, L., Zhu, F., Shao, L.: Ranet: Ranking attention network for fast video object segmentation. In: ICCV. pp. 3977–3986 (2019) 10
45. Wen, P., Yang, R., Xu, Q., Qian, C., Huang, Q., Cong, R., Si, J.: DMVOS: discriminative matching for real-time video object segmentation. In: ACMMM. pp. 2048–2056 (2020) 10
46. Xie, H., Yao, H., Zhou, S., Zhang, S., Sun, W.: Efficient regional memory network for video object segmentation. arXiv preprint arXiv:2103.12934 (2021) 2, 4, 10, 11
47. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.S.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018) 3, 9
48. Xu, X., Wang, J., Li, X., Lu, Y.: Reliable propagation-correction modulation for video object segmentation. In: AAAI. pp. 2946–2954 (2022) 10
49. Xu, Y., Fu, T., Yang, H., Lee, C.: Dynamic video segmentation network. In: CVPR. pp. 6556–6565 (2018) 3
50. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by foreground-background integration. In: ECCV. pp. 332–348 (2020) 2, 4, 10
51. Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. arXiv preprint arXiv:2106.02638 (2021) 4, 10, 11
52. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by multi-scale foreground-background integration. IEEE TPAMI (2021) 2
53. Zeng, Y., Zhang, P., Lin, Z.L., Zhang, J., Lu, H.: Towards high-resolution salient object detection. In: ICCV. pp. 7233–7242 (2019) 9
54. Zhang, P., Hu, L., Zhang, B., Pan, P.: Spatial constrained memory network for semi-supervised video object segmentation. CVPR Workshops (2020) 8
55. Zhou, Z., Ren, L., Xiong, P., Ji, Y., Wang, P., Fan, H., Liu, S.: Enhanced memory network for video segmentation. In: ICCV Workshops. pp. 689–692 (2019) 9