Query-Memory Re-Aggregation for Weakly-supervised Video Object Segmentation

Fanchao Lin[1], **Hongtao Xie**[1]*, **Yan Li**[1,2], **Yongdong Zhang**[1]
[1] School of Information Science and Technology, University of Science and Technology of China, Hefei, China
[2] Beijing Kuaishou Technology Co., Ltd., Beijing, China
lfc1995@mail.ustc.edu.cn, {htxie,zhyd73}@ustc.edu.cn, liyan@kuaishou.com

Abstract

Weakly-supervised video object segmentation (WVOS) is an emerging video task that can track and segment the target given a simple bounding box label. However, existing WVOS methods are still unsatisfied in either speed or accuracy, since they only use the exemplar frame to guide the prediction while they neglect the reference from other frames. To solve the problem, we propose a novel Re-Aggregation based framework, which uses feature matching to efficiently find the target and capture the temporal dependencies from multiple frames to guide the segmentation. Based on a two-stage structure, our framework builds an information-symmetric matching process to achieve robust aggregation. In each stage, we design a Query-Memory Aggregation (QMA) module to gather features from the past frames and make bidirectional aggregation to adaptively weight the aggregated features, which relieves the latent misguidance in unidirectional aggregation. To exploit the information from different aggregation stages, we propose a novel coarse-fine constraint by using the Cascaded Refinement Module (CRM) to combine the predictions from different stages and further boost the performance. Experimental results on three benchmarks show that our method achieves the state-of-the-art performance in WVOS (*e.g.*, an overall score of 84.7% on the DAVIS 2016 validation set).

Introduction

Video object segmentation (VOS) aims to track and give the pixel-wise identification of specific objects in a video sequence, which has many application scenarios like content-based interaction, video editing, video conference, and so on. With the prevalence of large-scale video-based datasets (Perazzi et al. 2016; Pont-Tuset et al. 2017; Xu et al. 2018), it has been a mainstream to solve the VOS using deep neural networks.

Most works in VOS can be divided into semi-supervised methods (Wang et al. 2019; Oh et al. 2019), unsupervised methods(Lu et al. 2019; Ventura et al. 2019) and weakly-supervised methods(Wang et al. 2019; Voigtlaender, Luiten, and Leibe 2019). As shown in Fig. 1, semi-supervised methods can generate accurate segmentation results given a concrete mask label in the exemplar frame.

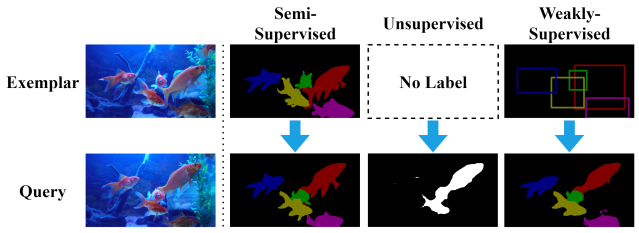

Figure 1: Visual comparison between semi-supervised, unsupervised and weakly-supervised VOS. The first row is the exemplar frame and corresponding target labels. The second row shows the query frame and corresponding segmentation results. Only the weakly-supervised VOS can tackle multiple targets with the simple labels.

However, the acquisition of the pixel-wise segmentation label is time-consuming and may be impractical in many cases. Unsupervised methods do not need any prior label and can make tracking and segmentation for the salient object, but they usually need to decide a main target in a video, leading to suboptimal accuracy and the restricted use in the cases with multiple targets.

In view of the limitations in both semi-supervised and unsupervised solutions, recent researchers tend to solve the VOS task in a weakly-supervised manner, which gives an easily obtained bounding box label as prior. In this way, users can conveniently mark the objects they want and the number of objects is not limited. SiamMask (Wang et al. 2019) uses the feature matching in a light-weighted network to make real-time tracking and segmentation. Though fast enough, its performance is not satisfactory as it only uses the exemplar frame as guidance. BoltVOS (Voigtlaender, Luiten, and Leibe 2019) designs a two-stage framework where the first stage tracks the bounding box and the second stage makes segmentation. Due to the utilization of conditional R-CNN (Ren et al. 2015) and extra segmentation network, this method achieves high accuracy while runs at a low speed.

Inspired by the recent temporal aggregation mechanism in video tasks, we use an aggregation based framework to take care of both efficiency and accuracy. The temporal aggregation usually calculates a pixel-wise similarity matrix

*Corresponding author

to match the query frame and past frames, then the features from the past frames can be efficiently aggregated according to the matching result. When applying such process in the weakly-supervised video object segmentation (WVOS) task, however, there exist the problem of asymmetric information, $i.e.$, the target information is known in the past frames but not provided in the query frame, which may cause wrong matching to the unexpected instances or the background noise. To solve this problem, we propose a two-stage Re-Aggregation framework to align the information from the query frame and past frames ($i.e.$, memory frames). Our framework gives the target prior to the query frame after the first stage, so that the query and memory features can form semantic-consistent pairs in the second stage to achieve more robust feature aggregation. As the aggregation based on the similarity matrix is a unidirectional process and lacks the mutual revalidation between the query frame and past frames, we expand the traditional aggregation and use a Bidirectional Channel Aggregation to bilaterally select the more discriminative channels and enhance the aggregated feature. To generate accurate predictions from the aggregated feature, we design a Cascade Refinement Module to combine and restrain the raw predictions and further improve the segmentation results. Experiments on three datasets demonstrate that our method achieves the state-of-the-art results among the weakly-supervised methods and is competitive with some semi-supervised methods while only using a bounding box prior.

Our main contributions are as follows:

- We propose a fast and accurate Query-Memory Re-Aggregation (QMRA) framework for the WVOS task, which achieves the state-of-the-art performance.
- We use two-stage feature aggregation to solve the information asymmetry in the aggregation process and conduct bidirectional aggregation in each stage to better gather the target-specific features from multiple frames.
- A Cascaded Refinement Module (CRM) is proposed to contact and constrain the information in different aggregation stages and optimize the final results.

Related Work

Temporal Feature Aggregation

Feature aggregation aims at combining the different-source features and taking the full advantage of information among them for better feature representation. In video tasks, the temporal feature aggregation is powerful to gather multi-frame information and make robust guidance.

TCENet (He et al. 2020) exploits temporal context information by the temporal aggregation for video object detection. AU-GACN (Xie et al. 2020) leverages the self-attention graph pooling to select useful information for expression recognition. In the visual tracking task, siamese-based approaches (Bertinetto et al. 2016; Sio et al. 2020) calculate the cross-correlation between the template and the search image to catch the target. For the unsupervised video object segmentation, COSNet (Lu et al. 2019) incorporates a global co-attention mechanism to get information from multiple reference frames. Among the semi-supervised video object segmentation methods, STMN (Oh et al. 2019) uses a Memory Read block to first calculate a pixel-wise similarity matrix between the query and memory frames and then aggregate the target features from the memory frames. As the memory frames contain the target information, the aggregated features thus can be target-aware.

There exist two potential problems when applying the aggregation process to the WVOS task: (1) The memory frames contain the target information while the query frame does not, which leads to information asymmetry during the calculate of the similarity matrix. (2) The aggregation is only made from the memory to the query, which is unidirectional and lacks the mutual perception and validation. To remedy these issues, we propose a two-stage Re-Aggregation framework to provide the target prior to the query frame and build a semantic-symmetric aggregation process. Moreover, we design a Query-Memory Aggregation module to achieve the bi-directional selection of the features between the query and memory.

Video Object Segmentation

The video object segmentation task can be applied in semi-supervised, unsupervised, and weakly-supervised forms, which are applicable in different scenarios. Semi-supervised VOS (SVOS) gives the target's initial segmentation label, which is a strong prior to understand the target. SVOS methods (Johnander et al. 2019; Wang et al. 2019; Oh et al. 2019; Seong, Hyun, and Kim 2020) can achieve excellent performance, but the need of the complex pixel-wise label limits their practical uses. Unsupervised VOS (UVOS) requires to predict the object masks automatically without any label. Lacking prior information, methods of this subject (Gu et al. 2020; Wang et al. 2019; Lu et al. 2019; Yang et al. 2019) usually need to judge a main object for prediction. Due to the non-label characteristic, it is not suitable to use the UVOS methods in the multi-object scenario.

Weakly-supervised VOS (WVOS), which provides a rough label ($i.e.$, a bounding box label in this paper) instead of a pixel-wise segmentation label, can be seen as an intermediate option between the SVOS and the UVOS. The WVOS methods are easy-to-use compared to the SVOS methods and are capable of segmenting multiple objects compared to the UVOS methods. Only a few works have been done for the WVOS. SiamMask (Wang et al. 2019) crops the object region using the box-level label and gets the cross-correlated features between the cropped region and the query image through siamese networks. It achieves real-time speed at the cost of suboptimal performance. BoltVOS (Voigtlaender, Luiten, and Leibe 2019) splits the WVOS task into two sub-tasks: the box-level tracking, and the segmentation of the bounding box. The use of Conditional R-CNN (Ren et al. 2015) in its framework results in a good performance and slow processing speed. These two methods can not make a good balance between speed and accuracy. In this paper, however, we explore a Re-Aggregation based framework, which is both fast and strong and can be trained in an end-to-end way.

Proposed Method

The definition of the WVOS in this paper is: given a video sequence $\mathbf{I} = \{I_t\}_{t=0}^{N}$, I_0 is the exemplar frame where the target bounding box label B_0 is provided, we are required to predict the targets' segmentation results $\mathbf{S} = \{S_t\}_{t=0}^{N}$. In our framework, we extra predict the target bounding boxes $\mathbf{B} = \{B_t\}_{t=1}^{N}$ at each frame except I_0. For the k-th frame I_k to be predicted ($i.e.$, the query frame), we further define the memory frames $\mathbf{I}_m$ which consist of I_0, I_{k-1}, and several extra frames which are sampled between them at a regular interval, $i.e.$, $\mathbf{I}_m = \{I_0, I_i, ..., I_{ni}, ..., I_{k-1}\}$, and define the corresponding memory bounding boxes $\mathbf{B}_m$ (ground-truth box in I_0 and predicted boxes in the other memory frames). We use the memory frames $\mathbf{I}_m$ and memory bounding boxes $\mathbf{B}_m$ to help predict the k-th segmentation result S_k.

The overall framework is shown in Fig. 2. We conduct the two-stage Re-Aggregation to generate the bounding box and segmentation result of the target sequentially. For each stage, the QMA module is used to gather robust target features. The CRM is applied to contact and refine the predictions from different stages.

Re-Aggregation Framework

The aggregation-based method aims to utilize the rich information in memory frames and give robust guidance to the prediction in the query frame. However, there is a problem of information asymmetry during such an aggregation process. That is, the features from the memory frames contain both the visual information from the images and the target information given by the labels and previous predictions, while the query frame does not know about the target. As a result, the pixel-wise matching between the query and memory is easy to be misled by the error target or the similar background, causing inferior aggregation results. In this paper, we propose a novel Re-Aggregation framework to tackle the information asymmetry problem by providing the target prior to the query frame and align the different-source features.

The Re-Aggregation framework consists of two aggregation stages. For the sake of meaning, here we redefine the query frame as I_q. The first stage takes the query frame I_q, the memory frames $\mathbf{I}_m$, and the memory bounding box maps $\mathbf{B}_m$ as input, calculating a similarity matrix $\mathbf{A}_1$ to aggregate the pixel-wise features and predict a rough target mask M_q in the query frame:

$$\mathbf{A}_1 = \mathbf{F}_m^T \mathbf{F}_q = [f_m(\mathbf{I}_m, \mathbf{B}_m)]^T f_q(I_q) \tag{1}$$

$$M_q = \varphi_1(\mathbf{A}_1, \mathbf{F}_m, \mathbf{F}_q) \tag{2}$$

here f_m is the memory encoder which takes 4-channel input (concatenated RGB frame and bounding box map) and generates the memory feature $\mathbf{F}_m$, f_q is the query encoder which takes the 3-channel image as input and generates the query feature $\mathbf{F}_q$. φ denotes the function of the feature aggregation and the decoder network. As can be seen in Eq. 1, the feature encoding procedure is asymmetric between the query and memory frames in the first stage, thus can not generate a robust similarity matrix $\mathbf{A}_1$ to guide accurate predictions. However, we can treat the result of M_q as a rough target indicator. In the second stage, M_q is used as a basic prior to provide the position and scale information of the potential target and generate a more detailed segmentation result S_q:

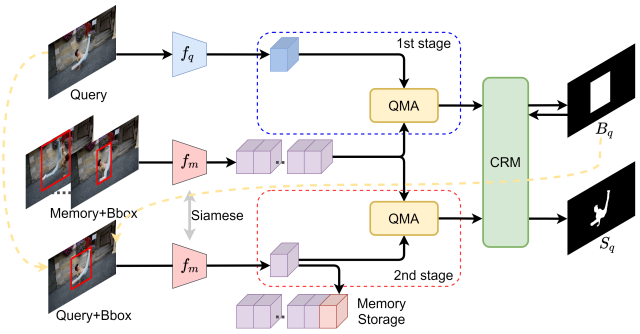

Figure 2: Query-Memory Re-Aggregation Framework. The first aggregation stage generates a bounding box map for the target in the query frame. The second aggregation stage considers the bounding box prior and predicts a concrete segmentation mask.

$$\mathbf{A}_2 = \mathbf{F}_m^T \mathbf{F}_q^r = [f_m(\mathbf{I}_m, \mathbf{B}_m)]^T f_q^r(I_q, M_q) \tag{3}$$

$$S_q = \varphi_2(\mathbf{A}_2, \mathbf{F}_m, \mathbf{F}_q^r) \tag{4}$$

here M_q is concatenated with the query image I_q and input to another query encoder f_q^r to generate the new query feature $\mathbf{F}_q^r$. We can see from Eq. 3 that both the query and memory frames are encoded with the target information, then the cross-correlation between $\mathbf{F}_m$ and $\mathbf{F}_q^r$ is information-symmetric to generate a more robust similarity matrix $\mathbf{A}_2$. If we further constrain that $f_q^r = f_m$, then it forms a siamese encoder structure between the query and the memory frames:

$$\mathbf{A}_2 = [f_m(\mathbf{I}_m, \mathbf{B}_m)]^T f_m(I_q, M_q) \tag{5}$$

In this way, the query and memory features are generated from the consistent input form and the same encode operation, making them semantically aligned for the feature matching and aggregation. What's more, the prior masks provided in both query and memory frames can make the aggregation process pay more attention to the latent target areas from the beginning. Our extensive experiments demonstrate the effectiveness of the Re-Aggregation framework and its superiority over one-stage aggregation. It is worth noting that the predicted target mask M_q in Eq. 1 can be various forms to initially indicate the target. For the WVOS, we set M_q as the target bounding box map ($i.e.$, $M_q = B_q$), as shown in Fig. 2. This naturally forms a track-segment pipeline, with the first stage aggregation to track the object and the second stage to give the pixel-level segmentation.

Considering the efficiency, the encoding process of the query frame in the second stage is designed to play two roles, not only provides the query feature for aggregation but also generates the memory feature for the following frames. That is, the encoded feature $\mathbf{F}_q^r = f_m(I_q, M_q)$ is added to the memory features and guides the predictions in the next

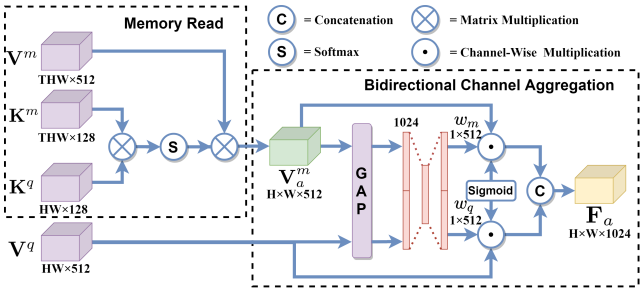

Figure 3: Our Query-Memory Aggregation (QMA) module. The Memory Read block gathers features from the memory frames. The Bidirectional Channel Aggregation combines the aggregated memory feature and the query feature to achieve mutual perception for the target.

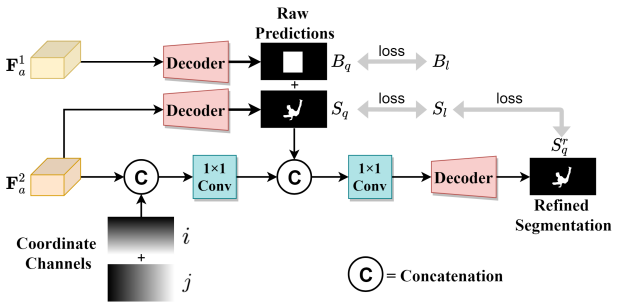

Figure 4: The Cascaded Refinement Module (CRM) used after the raw predictions. $\mathbf{F}_a^1$ and $\mathbf{F}_a^2$ are the aggregated features at different stages.

frame. In that case, the memory features are constantly replenished and we save them in storage so that the memory frames do not need to be encoded repeatedly, which greatly accelerates our method.

Query-Memory Aggregation

The guidance from the past frames plays a key role in obtaining target-specific information. Previous works of the WVOS (Wang et al. 2019; Voigtlaender, Luiten, and Leibe 2019) only utilize the information given in the exemplar frame, easy to cause the performance drop when the large deformation and displacement appear. In order to take full advantage of the video sequence, we propose a Query-Memory Aggregation (QMA) module to utilize information from multiple frames and highlight the target-related features in an efficient and robust way, as shown in Fig. 3.

Suppose there are T memory frames, following (Oh et al. 2019), we use the 1×1 convolution layers to transform the memory features into memory key $\mathbf{K}^m \in \mathbb{R}^{T\times H\times W\times 128}$ and memory value $\mathbf{V}^m \in \mathbb{R}^{T\times H\times W\times 512}$ and transform the query feature into query key $\mathbf{K}^q \in \mathbb{R}^{H\times W\times 128}$ and query value $\mathbf{V}^q \in \mathbb{R}^{H\times W\times 512}$. Here 128 and 512 are the number of channels; H and W are respectively the height and width of the feature maps.

In our QMA module, we first use a Memory Read block (Oh et al. 2019) to gather the pixel-wise memory features:

$$\mathbf{V}_a^m = \mathrm{S}^{\mathrm{r}}[\mathbf{K}^q(\mathbf{K}^m)^T]\mathbf{V}^m \tag{6}$$

where $\mathrm{S}^{\mathrm{r}}[\mathbf{K}^q(\mathbf{K}^m)^T]$ is a $HW \times THW$ similarity matrix between each location of the query and memory features and $S^r[\cdot]$ denotes the row-wise softmax used to normalize the similarity matrix. Under our Re-Aggregation design, the similarity matrix is generated from asymmetric information in the first aggregation stage. While in the second stage, different from (Oh et al. 2019), the query bounding box is provided and the similarity matrix is calculated according to features with aligned semantics. Thus the feature aggregation process considers both visual and target information, which is more discriminative.

The Memory Read module aggregates the pixel-wise semantics from the memory features according to the degrees of similarity. However, Memory Read is a unidirectional aggregation ($i.e.$ from memory to query) and lacks the bilateral validation. The bidirectional attention mechanism (Lu et al. 2019; Liu et al. 2019) is demonstrated to help two modules enhance each other and the channel-wise attention (Hu, Shen, and Sun 2018; Min et al. 2020) can adaptively improve the feature representation. In our QMA module, we further conduct a Bidirectional Channel Aggregation (BCA) to achieve the mutual perception between the query and memory features, which can be formulated as follows:

$$\omega_q = f_{mq}\left(\mathbf{V}^q, \mathbf{V}_a^m\right) \tag{7}$$

$$\omega_m = f_{qm}\left(\mathbf{V}^q, \mathbf{V}_a^m\right) \tag{8}$$

$$\mathbf{F}_a = \{\omega_q \cdot \mathbf{V}^q, \omega_m \cdot \mathbf{V}_a^m\} \tag{9}$$

where ω_q and ω_m are the weight vectors, $\cdot$ means channel-wise multiplication and $\{\cdot,\cdot\}$ means concatenation. ω_q and ω_m combine the query and memory features to adaptively generate a weight for each feature channel. The details are shown in Fig. 3. Global Average Pooling (GAP) is applied on $\mathbf{V}_a^m$ and $\mathbf{V}^q$ to get two feature vectors of length 512. The feature vectors are concatenated and passed through a linear unit which consists of two fully-connected layers with a bottleneck. The output of the linear unit is normalized by a sigmoid function and divided into two weight vectors. Then we use the jointly predicted vectors to re-weight $\mathbf{V}^q$ and $\mathbf{V}_a^m$ in the channel direction and concatenate the results to form the aggregated feature $\mathbf{F}_a$.

In the second stage of our framework, both $\mathbf{V}^q$ and $\mathbf{V}_a^m$ contain the target information from the bounding boxes and the bidirectional channel-wise aggregation can help to select the co-occurrence target features and inhibit the influence of the background noise. Experimental results prove that such a design can help the network gather the multi-frame target information and boost the performance.

Cascaded Refinement and Prediction

In our framework, the raw predictions of target bounding box B_q and segmentation mask S_q are generated separately in different stages, which means they lack the mutual contact and constraint to each other. To link the predicted bounding box and segmentation results, we design a Cascaded Refinement Module (CRM), as shown in Fig. 4. Inspired by (Cho

et al. 2020), a CoordConv (Liu et al. 2018) layer is first used to combine the aggregated feature $\mathbf{F}_a$ with the coordinate information. In the CoordConv layer, two coordinate channels are created. The i coordinate channel is a matrix with its first row filled with 0's, its second row with 1's, its third row with 2's, etc. The j coordinate channel is similar but in the column direction. A linear scaling for both i and j coordinate channels is applied to make the values fall in the range [-1, 1]. Then we sequentially merge the aggregated feature with the i, j coordinate channels, and the predicted B_q and S_q through the channel-wise concatenation and 1×1 convolution layers. The new aggregated feature is input to a decoder and predicts the refined segmentation mask S_q^r. In this way, the new aggregated feature contains the coordinate information which can help the network be more location-aware. Besides, the two-stage predictions can make a supplement to generate a more consistent segmentation.

To generate each prediction (B_q, S_q, S_q^r) from the aggregated features, we simply use the decoder network follows (Wug Oh et al. 2018). The loss function of the whole framework is:

$$L = l(B_q, B_l) + l(S_q, S_l) + l(S_q^r, S_l) \tag{10}$$

where l is a combination of the dice loss and the cross-entropy loss, and the weight of dice loss is set to 0.1 by experience. B_l and S_l are the corresponding labels.

Experiments

In this section, we evaluate the proposed method and compare it to state-of-the-art methods from two aspects: performance and running speed.

Implementation Details

We take a two-stage training procedure based on static data and video data, respectively. For the pre-training on static data, we generate pairs of simulative video frames from the salient object segmentation datasets, $i.e.$, DUTS (Wang et al. 2017), HKU-IS (Li and Yu 2015), MSRA (Cheng et al. 2014), and SOC (Fan et al. 2018). Every image-mask pair in the raw static datasets is augmented into three pairs using thin-plate splines transformations, rotations, and random cropping to obtain the diversity. After the pre-training on the static data, we continue to train the network using the training set of YouTube-VOS (Xu et al. 2018), which is a real-world VOS dataset. We randomly sample three temporally ordered frames in a video and the maximum frame interval is 5. During both training processes, we set three image-label pairs as a group. The first two pairs are used as the memory frames with their bounding box labels, and the last pair is set as the query frame to be predicted. For the encoders in our framework, both the query encoder f_q and the memory encoder f_m use the ResNet50 (He et al. 2016) till the 4-th stage as the backbone, but f_m adds extra filters in the input layer so that it can take 4 channels (RGB frame and a bounding box map) as input. We train our network using the Adam algorithm with a fixed learning rate of 1e-5 on four GTX 1080Ti GPUs, and the batch size is 16.

During the inference, only the initial bounding box label is given and the prediction is made in a propagation-like way. When multiple targets exist, we predict each target individually and use a softmax function to determine the most likely category each pixel belongs to. Our method is evaluated on a computer with a single V100 GPU.

	$\mathcal{L}$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$	FPS
PCSA	N	77.2	87.8	77.4	84.4	110.0
AGS	N	79.7	91.1	77.4	85.8	-
COSNet	N	80.5	94.0	79.4	90.4	-
AD-Net	N	81.7	-	80.5	-	-
MATNet	N	**82.4**	**94.5**	**80.7**	**90.2**	-
FEELVOS	S	81.1	90.5	82.2	86.6	2.2
RGMP	S	81.5	91.7	82.0	90.8	2.2
A-GAME	S	82.0	-	82.2	-	14.3
SAT	S	82.6	-	83.6	-	39.0
RANet	S	85.5	**97.2**	85.4	**94.9**	30.3
STMN	S	88.7	-	89.9	-	6.3
KMN	S	**89.5**	-	**91.5**	-	8.3
SiamMask	B	71.7	86.8	67.8	79.8	33.6
BoltVOS	B	78.1	-	81.2	-	1.4
QMRA (Ours)	B	**84.7**	**96.7**	**87.1**	**95.6**	12.1

Table 1: Comparison with recent state-of-the-art methods on DAVIS 2016 validation set.

	$\mathcal{L}$	YouTube-VOS		DAVIS17	
		$\mathcal{G}$	FPS	$\mathcal{J}\&F$	FPS
RVOS	N	**33.7**	-	-	-
RGMP	S	53.8	3.8	66.7	3.6
FEELVOS	S	-	-	69.1	2.0
A-GAME	S	66.1	-	70.0	-
TVOS	S	67.8	37.0	72.3	37.0
FRTM	S	72.1	21.9	76.7	21.9
STMN	S	79.4	-	81.7	-
KMN	S	**81.4**	-	**82.8**	-
SiamMask	B	52.8	16.7*	54.3	16.7*
BoltVOS	B	65.7	0.7	**71.9**	0.7
LWL	B	**70.2**	3.0*	70.6	3.0*
QMRA (Ours)	B	67.6	6.4	**71.9**	6.3

Table 2: Comparison with recent state-of-the-art methods on YouTube-VOS validation set and DAVIS 2017 validation set. * means the value is extrapolated from DAVIS 2016.

Datasets and Metrics for Evaluation

We conduct experiments on three public datasets: the single-object DAVIS 2016 (Perazzi et al. 2016) dataset and the multi-object DAVIS 2017 (Pont-Tuset et al. 2017) and YouTube-VOS (Xu et al. 2018) datasets. Region similarity metric $\mathcal{J}$ and boundary accuracy metric $\mathcal{F}$ are used as the performance measures. Frame-per-second (FPS) during inference is the measure for running speed.

DAVIS 2016 contains 50 videos, which are divided into a training set (30videos) and a validation set (20 videos). The exemplar frame in this dataset always appears in the first frame of a video.

Network Variants	DAVIS 2016		YouTube-VOS	
	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{J}_s$	$\mathcal{J}_u$
Re-Aggregation Framework				
1stage	81.4	92.2	67.4	53.7
2stage-NoAlign	82.8	94.8	69.2	55.3
2stage-NoSiam	82.4	92.0	69.5	55.7
2stage (ours)	84.2	95.6	70.3	56.7
Query-Memory Aggregation Module				
w/o QMA	72.2	81.5	53.7	32.2
w/o BCA	81.8	93.0	69.7	55.3
w/ QMA (ours)	84.2	95.6	70.3	56.7
Cascaded Refinement Module				
w/o CRM	82.2	91.4	69.8	55.5
CRM w/o CL	82.8	94.2	70.3	55.7
w/ CRM (ours)	84.2	95.6	70.3	56.7
Other Variations				
w/o EF	82.5	93.2	69.7	55.5
w/ EF (ours)	84.2	95.6	70.3	56.7
ours+COCO	84.7	96.7	71.2	58.1

Table 3: Overall ablation studies on key parts of our framework.

DAVIS 2017 is an extended dataset of DAVIS 2016, which has 60 videos for training and 30 videos for validation with multiple targets per video. For both DAVIS datasets, the mean and recall of $\mathcal{J}$ and $\mathcal{F}$ are calculated, denoted as $\mathcal{J}_m$, $\mathcal{J}_r$, $\mathcal{F}_m$, and $\mathcal{F}_r$, respectively. $\mathcal{J}\&\mathcal{F}$ is the average score of $\mathcal{J}_m$ and $\mathcal{F}_m$.

YouTube-VOS is a large-scale dataset consists of 3471 training videos and 474 validation videos. In this dataset, the measures of $\mathcal{J}$ and $\mathcal{F}$ are separately calculated for the seen ($\mathcal{J}_s$, $\mathcal{F}_s$) and unseen ($\mathcal{J}_u$, $\mathcal{F}_u$) object classes during the training, and the overall score $\mathcal{G}$ is the average of $\mathcal{J}$ and $\mathcal{F}$.

Comparison with State-of-the-Art Methods

In this part, we make the overall comparisons between the UVOS, SVOS, and WVOS methods to show their differences. The UVOS methods include PCSA (Gu et al. 2020), AGS (Wang et al. 2019), COSNet (Lu et al. 2019), AD-Net (Yang et al. 2019), MATNet (Zhou et al. 2020) and RVOS (Ventura et al. 2019). The SVOS methods include FEELVOS (Voigtlaender et al. 2019), RGMP (Wug Oh et al. 2018), A-GAME (Johnander et al. 2019), SAT (Chen et al. 2020), RANet (Wang et al. 2019), STMN (Oh et al. 2019), KMN (Seong, Hyun, and Kim 2020), TVOS (Zhang et al. 2020), FRTM (Robinson et al. 2020). Specific for the WVOS, we compare the performance of our method with other three state-of-the-art methods which also use the bounding box as a prior: SiamMask (Wang et al. 2019), BoltVOS (Voigtlaender, Luiten, and Leibe 2019) and LWL (Bhat et al. 2020). SiamMask can run at real-time speed due to a light-weighted framework but it can not well handle the large displacement. BoltVOS utilizes the RCNN and DeepLabv3+ modules to achieve robust predictions in the multi-object scenario but also lead to slow processing speed. LWL builds a few-shot learner based on the exemplar frame and applies steepest descent iterations to find an approximate solution. It requires multiple iterations in each frame, which limits its practical efficiency. The quantitative results are shown in Table 1 and Table 2, where N, S, and B denote the use of no label, segmentation label, and bounding box label in the exemplar frame, respectively.

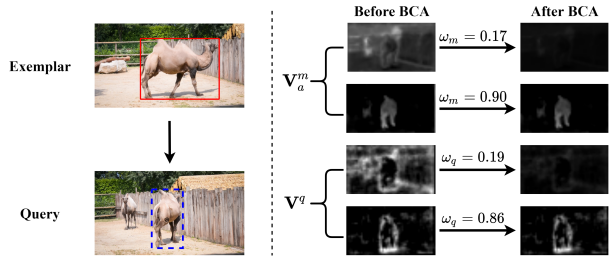

Figure 5: Visualization of the feature maps before and after the Bidirectional Channel Aggregation (BCA). The numbers on the arrows indicate the weights for channels.

Single Object. Table 1 shows the evaluation results of the single-object case on the DAVIS 2016 validation set. Our method performs comparable to some recent SVOS methods. Among the WVOS methods, SiamMask (Wang et al. 2019) runs at 33 FPS, but the recall is relatively low (86.8% on $\mathcal{J}_r$) because it only uses the guidance of the exemplar frame. BoltVOS (Voigtlaender, Luiten, and Leibe 2019) achieves better performance than SiamMask, but it costs too much processing time. In terms of speed, our method is slower than SiamMask but much faster than BoltVOS because we apply the more efficient aggregation-based guidance. For the performance evaluation, our method surpasses SiamMask by 13.0% and BoltVOS by 6.6% on $\mathcal{J}_m$. Besides, we achieve $\mathcal{J}_r$ of 96.7% and $\mathcal{F}_r$ of 95.6%, which shows that our method is powerful to catch the object.

Multiple Objects. Evaluation results for the multi-object cases on the YouTube-VOS validation set and DAVIS 2017 validation set are reported in Table 2. The only UVOS method under this scenario is RVOS (Ventura et al. 2019), which only achieves an overall score of 33.7% in the YouTube-VOS validation set. This demonstrates that the UVOS methods are not suitable for the multi-object task. The proposed QMRA also achieves competitive performance against some SVOS methods. Among the WVOS methods, our method outperforms SiamMask (Wang et al. 2019) by 14.8% and BoltVOS (Voigtlaender, Luiten, and Leibe 2019) by 1.9% in the YouTube-VOS validation set. The performance of our method is inferior to the most recent LWL (Bhat et al. 2020) in this dataset, but our running speed is double of it. In the DAVIS 2017 validation set, our method surpasses SiamMask and LWL, sharing a best performance with BoltVOS. Note that our method runs about 8 times faster than BoltVOS in the same V100 GPU, demonstrating its superior efficiency.

Ablation Studies and Analysis

Studies on Re-Aggregation framework. Our Re-Aggregation framework aims at providing the target prior

Figure 6: Qualitative results of our method on DAVIS 2016 (1st and 2nd rows) and YouTube-VOS (3rd and 4th rows). The first column shows the exemplar frames where the bounding box labels are marked.

to the query and forming the semantic-symmetric feature aggregation. Experiments are conducted in the first part of Table 3. 1stage means we only use one-stage feature aggregation to predict the segmentation. 2stage-NoAlign denotes that we do not keep a consistent input form (RGB frame and bounding box map) between the query and memory in the second stage, but predict the segmentation maps in both stages instead. 2stage-NoSiam indicates that we use an individual encoder f_q^r instead of f_m to encode I_q and M_q in Eqn. 3. Our two-stage Re-Aggregation design is prominently superior to the one-stage framework (increments of 2.9% and 3.0% on $\mathcal{J}_s$ and $\mathcal{J}_u$, respectively). The performance drops of 2stage-NoAlign and 2stage-NoSiam comparing to 2stage (Ours) prove that the consistent input form and the use of the siamese encoder are both important to build the semantic-symmetric feature aggregation.

Studies on Query-Memory Aggregation Module. The effect of Query-Memory Aggregation (QMA) is reported in the second part of Table 3. w/o QMA means the network simply concatenates the query feature and memory features without using an aggregation module. Applying the Memory Read module individually (w/o BCA) makes an improvement of 9.6% on metric $\mathcal{J}_m$, which shows that the similarity measure is strong guidance for feature aggregation. Adding BCA (from w/o BCA to w/ QMA) further brings a significant improvement on $\mathcal{J}_m$ (from 81.8% to 84.2%). Notably, BCA increases $\mathcal{J}_s$ by 0.6% while increases $\mathcal{J}_u$ by 1.4%, indicating that BCA can enhance the generalization of the network and make it more robust for unseen categories. Fig. 5 visualizes the feature maps before and after the use of BCA. It can be seen that BCA tends to suppress the channels with high responses on the error target or the background and keep the channels having distinguishing target features.

Studies on Cascaded Refinement Module. We apply a Cascaded Refinement Module (CRM) to contact and refine the two-stage predictions. Using the CRM without the CoordConv layer (CRM w/o CL) boosts the metric $\mathcal{J}_r$ by 2.8%. This result indicates that the extra refinement process can tie the predictions at different stages and increase the recall. Adding the CoordConv layer (CRM (ours)) narrows the gap between $\mathcal{J}_s$ and $\mathcal{J}_u$ and brings an improvement of 1.4% on $\mathcal{J}_r$, which shows that the provided coordinate information helps to catch the target and improve the robustness.

Studies on Other Variations. The extra memory frames are sampled between the exemplar frame and the previous frame and the sampling interval is set to 10, 10, and 5 for the evaluation of DAVIS 2016, DAVIS 2017, and YouTube-VOS datasets, respectively. The use of extra memory frames (w/ EF) can boost the $\mathcal{J}_r$ by 2.4%, which indicates that the introduction of more frames is helpful to learn about the target information. Adding COCO (Lin et al. 2014) dataset into pre-training further boosts the performance, especially for the unseen categories (1.4% on $\mathcal{J}_u$).

Qualitative Results. The qualitative results of our method are shown in Fig. 6. The proposed QMRA can catch the fast-moving target (1st row), discriminate similar instances (2nd row), and handle the cases where the targets close to each other (3rd row). However, some failure segmentations still exist around the boundaries of adjacent targets (4th row) when the given box-level labels have too much overlap.

Conclusion

In this work, we propose a fast and accurate QMRA framework, promoting the task of weakly-supervised VOS towards practical use. With a two-stage structure, our method tackles the problem of asymmetric information between the query and memory frames during the feature aggregation. To gather robust target features, we expand the Memory Read block and use a channel-wise aggregation to make the bidirectional feature selection between the query and memory features. The proposed Cascade Refinement Module can contact and restrain the raw predictions and further boost the performance. Moving forward, we are going to develop our Re-Aggregation method to other related tasks.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2017YFC0820600), the National Nature Science Foundation of China (61525206, 62022076, U1936210), the Youth Innovation Promotion Association Chinese Academy of Sciences (2017209).

We also acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, 850–865. Springer.

Bhat, G.; Lawin, F. J.; Danelljan, M.; Robinson, A.; Felsberg, M.; Gool, L. V.; and Timofte, R. 2020. Learning What to Learn for Video Object Segmentation. *arXiv preprint arXiv:2003.11540* .

Chen, X.; Li, Z.; Yuan, Y.; Yu, G.; Shen, J.; and Qi, D. 2020. State-Aware Tracker for Real-Time Video Object Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9384–9393.

Cheng, M.-M.; Mitra, N. J.; Huang, X.; Torr, P. H.; and Hu, S.-M. 2014. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3): 569–582.

Cho, S.; Cho, M.; Chung, T.-y.; Lee, H.; and Lee, S. 2020. CRVOS: Clue Refining Network for Video Object Segmentation. *arXiv preprint arXiv:2002.03651* .

Fan, D.-P.; Cheng, M.-M.; Liu, J.-J.; Gao, S.-H.; Hou, Q.; and Borji, A. 2018. Salient objects in clutter: Bringing salient object detection to the foreground. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 186–202.

Gu, Y.; Wang, L.; Wang, Z.; Liu, Y.; Cheng, M.-M.; and Lu, S.-P. 2020. Pyramid Constrained Self-Attention Network for Fast Video Salient Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

He, F.; Gao, N.; Li, Q.; Du, S.; Zhao, X.; and Huang, K. 2020. Temporal Context Enhanced Feature Aggregation for Video Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Johnander, J.; Danelljan, M.; Brissman, E.; Khan, F. S.; and Felsberg, M. 2019. A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8953–8962.

Li, G.; and Yu, Y. 2015. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5455–5463.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision – ECCV 2014*, 740–755. Cham: Springer International Publishing. ISBN 978-3-319-10602-1.

Liu, C.; Xie, H.; Zha, Z.-J.; Yu, L.; Chen, Z.; and Zhang, Y. 2019. Bidirectional attention-recognition model for fine-grained object classification. *IEEE Transactions on Multimedia* .

Liu, R.; Lehman, J.; Molino, P.; Petroski Such, F.; Frank, E.; Sergeev, A.; and Yosinski, J. 2018. An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution. In *Advances in Neural Information Processing Systems*.

Lu, X.; Wang, W.; Ma, C.; Shen, J.; Shao, L.; and Porikli, F. 2019. See More, Know More: Unsupervised Video Object Segmentation with Co-Attention Siamese Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3623–3632.

Min, S.; Yao, H.; Xie, H.; Zha, Z.-J.; and Zhang, Y. 2020. Multi-Objective Matrix Normalization for Fine-Grained Visual Recognition. *IEEE Transactions on Image Processing* 29: 4996–5009.

Oh, S. W.; Lee, J.-Y.; Xu, N.; and Kim, S. J. 2019. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 9226–9235.

Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; and Sorkine-Hornung, A. 2016. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *Computer Vision and Pattern Recognition*.

Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 DAVIS Challenge on Video Object Segmentation. *arXiv:1704.00675* .

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Robinson, A.; Lawin, F. J.; Danelljan, M.; Khan, F. S.; and Felsberg, M. 2020. Learning Fast and Robust Target Models for Video Object Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7406–7415.

Seong, H.; Hyun, J.; and Kim, E. 2020. Kernelized Memory Network for Video Object Segmentation. *arXiv preprint arXiv:2007.08270* .

Sio, C. H.; Ma, Y.-J.; Shuai, H.-H.; Chen, J.-C.; and Cheng, W.-H. 2020. S2SiamFC: Self-supervised Fully Convolutional Siamese Network for Visual Tracking. In *Proceedings*

of the 28th ACM International Conference on Multimedia, 1948–1957.

Ventura, C.; Bellver, M.; Girbau, A.; Salvador, A.; Marques, F.; and Giro-i Nieto, X. 2019. Rvos: End-to-end recurrent network for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5277–5286.

Voigtlaender, P.; Chai, Y.; Schroff, F.; Adam, H.; Leibe, B.; and Chen, L.-C. 2019. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9481–9490.

Voigtlaender, P.; Luiten, J.; and Leibe, B. 2019. BoLTVOS: Box-Level Tracking for Video Object Segmentation. *arXiv preprint arXiv:1904.04552* .

Wang, L.; Lu, H.; Wang, Y.; Feng, M.; Wang, D.; Yin, B.; and Ruan, X. 2017. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 136–145.

Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; and Torr, P. H. 2019. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1328–1338.

Wang, W.; Song, H.; Zhao, S.; Shen, J.; Zhao, S.; Hoi, S. C. H.; and Ling, H. 2019. Learning Unsupervised Video Object Segmentation Through Visual Attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3059–3069.

Wang, Z.; Xu, J.; Liu, L.; Zhu, F.; and Shao, L. 2019. RANet: Ranking Attention Network for Fast Video Object Segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*.

Wug Oh, S.; Lee, J.-Y.; Sunkavalli, K.; and Joo Kim, S. 2018. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7376–7385.

Xie, H.-X.; Lo, L.; Shuai, H.-H.; and Cheng, W.-H. 2020. AU-assisted Graph Attention Convolutional Network for Micro-Expression Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2871–2880.

Xu, N.; Yang, L.; Fan, Y.; Yang, J.; Yue, D.; Liang, Y.; Price, B.; Cohen, S.; and Huang, T. 2018. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 585–601.

Yang, Z.; Wang, Q.; Bertinetto, L.; Hu, W.; Bai, S.; and Torr, P. H. S. 2019. Anchor Diffusion for Unsupervised Video Object Segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*.

Zhang, Y.; Wu, Z.; Peng, H.; and Lin, S. 2020. A Transductive Approach for Video Object Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6949–6958.

Zhou, T.; Wang, S.; Zhou, Y.; Yao, Y.; Li, J.; and Shao, L. 2020. Motion-Attentive Transition for Zero-Shot Video Object Segmentation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 13066–13073.