Memory Aggregation Networks for
Efficient Interactive Video Object Segmentation

Jiaxu Miao[1,2]   Yunchao Wei[2]   Yi Yang[2†]

[1]Baidu Research   [2] ReLER, University of Technology Sydney

jiaxu.miao@student.uts.edu.au, {yunchao.wei, yi.yang}@uts.edu.au

Abstract

Interactive video object segmentation (iVOS) aims at efficiently harvesting high-quality segmentation masks of the target object in a video with user interactions. Most previous state-of-the-arts tackle the iVOS with two independent networks for conducting user interaction and temporal propagation, respectively, leading to inefficiencies during the inference stage. In this work, we propose a unified framework, named Memory Aggregation Networks (MA-Net), to address the challenging iVOS in a more efficient way. Our MA-Net integrates the interaction and the propagation operations into a single network, which significantly promotes the efficiency of iVOS in the scheme of multi-round interactions. More importantly, we propose a simple yet effective memory aggregation mechanism to record the informative knowledge from the previous interaction rounds, improving the robustness in discovering challenging objects of interest greatly. We conduct extensive experiments on the validation set of DAVIS Challenge 2018 benchmark. In particular, our MA-Net achieves the J@60 score of 76.1% without any bells and whistles, outperforming the state-of-the-arts with more than 2.7%.

1. Introduction

Video object segmentation (VOS) aims at separating a foreground object from a video sequence and can benefit many important applications, including video editing, scene understanding, and self-driving cars. Most existing VOS approaches can be roughly divided into two settings: *unsupervised* (no manual annotation) and *semi-supervised* (give the annotation at the first frame). However, these two settings have their own limitations and are not realistic in practice: 1) unsupervised methods have no guiding signal for the user to select the object of interest, which
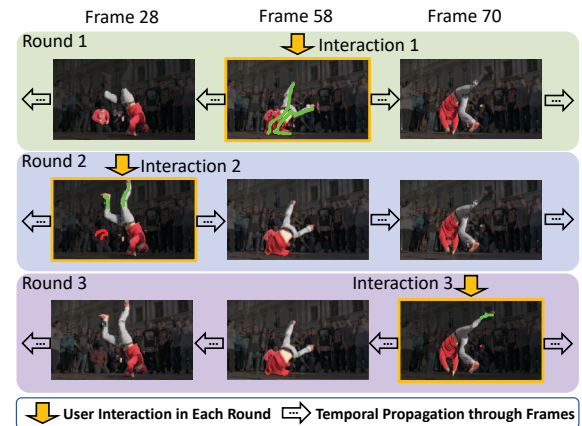

Figure 1. Round-based iVOS. The mask of the target object is generated by user annotations at one frame (*e.g.* green scribbles at frame 58), and the computed mask is propagated to generate the masks for the entire video. The user can refine the segmentation masks by repeatedly providing annotations on the false negative and false positive areas (*e.g.* green and red scribbles at frame 28).

is problematic especially for the multiple-object case; 2) semi-supervised methods need a fully annotated mask of the first frame, which is tedious to acquire (around 79 seconds per instance) [6]. Furthermore, for both two schemes, users have no chance to correct those low-quality segments to meet their requirements.

Interactive video object segmentation (iVOS) overcomes the above-mentioned limitations by providing a user-friendly annotation form, *e.g.*, scribbles. In this scheme, users can gradually refine the outputs by drawing scribbles on the falsely predicted regions. Previous iVOS methods [29, 25, 1] utilize a rotoscoping procedure [4, 15], where a user sequentially processes a video frame-by-frame. These methods are inefficient due to requiring a lot of user interactions at each frame.

Recently, Caelles *et al*. [6] propose a *round-based interaction* scheme, as shown in Fig. 1. In this setting, users firstly draw scribbles on the target objects at one selected frame, and an algorithm is then employed to compute the segmentation masks for all video frames with temporal

---

[†]Corresponding author.

[‡]Part of this work was done when Jiaxu Miao was an intern at Baidu Research.

propagation. The procedures of user annotation and mask segmentation are repeated until acceptable results are obtained. Such a *round-based interaction* scheme is more efficient since it requires fewer user annotations (only a few scribbles at one frame per round). Besides, it is flexible for users to control the quality of segmentation masks, since more rounds of user interactions will guarantee more accurate segmentation results.

In this paper, we explore how to build an efficient interactive system to tackle the iVOS problem under the round-based interaction setting. While some recent deep learning based methods [21, 12, 20, 3, 6] have been proposed to deal with the round-based iVOS, there are several limitations: 1) the user interaction and the temporal propagation are usually processed by two independent networks [12, 3]; 2) the whole neural network has to start a new feed-forward computation in each interaction round [21], or needs post-processing [20] to make a further refinement, which is time-consuming; 3) only the outputs of latest round are utilized to refine the segmentation results, while the informative multi-round interactions are usually ignored [12].

Considering these limitations, we propose a **unified**, **efficient**, and **accurate** framework named Memory Aggregation Networks (MA-Net) to deal with the iVOS in a more elegant and effective manner. Concretely, our MA-Net integrates the interaction network and propagation network into a unified pixel embedding learning framework by sharing the same backbone. In this way, after extracting the pixel embedding with the shared backbone, the MA-Net adopts two "shallow" convolutional segmentation heads to predict the object segments of the scribble-labeled frame and all the other frames, respectively. Under the round-based iVOS scheme, we only need to extract the pixel embedding of all the frames in the first round. In all the following rounds, these extracted embedding can be simply applied to make a further refinement with two "shallow" segmentation heads, resulting in our MA-Net much faster than previous methods. More importantly, we propose a simple yet effective memory aggregation mechanism, which is used to record informative knowledge of the user's interactions and the predicted masks during the previous interaction rounds. Such aggregated information makes the MA-Net robust to the target instances with a wide variety of appearances, improving the accuracy of our model greatly.

Our MA-Net is quantitatively evaluated on the interactive benchmark at the DAVIS Challenge 2018 [6]. On the DAVIS validation set, our MA-Net achieves the J@60 score of 76.1% without any bells and whistles, such as introducing additional optical flow information [12] or applying time-consuming CRF for post-processing [20, 14]. In addition, our MA-Net can accomplish 7-round interactions within 60 seconds, which is more efficient than the state of the art one [21] of 5-round interactions within 60 seconds.

2. Related Work

Unsupervised Video Object Segmentation. Unsupervised VOS does not need any user annotations. Most unsupervised segmentation models [26, 30] learn to automatically segment visually salient objects based on the motion information or the appearance information. The limitation of unsupervised VOS is that users cannot select the object of interest.

Semi-supervised Video Object Segmentation. Semi-supervised VOS employs the full annotation of the first frame to select the objects of interest. Many semi-supervised VOS methods [8, 13, 27, 32, 22, 34, 5, 28, 19, 35, 36] have been proposed and achieve good performance.

Some semi-supervised VOS approaches [5, 28, 16, 18] rely on fine-tuning using the first frame annotation at test time. For instance, OSVOS [5] employs a convolutional neural network pre-trained for foreground-background segmentation and fine-tunes the model using first-frame ground truth when testing. OnAVOS[28] and OSVOS-S [19] further improve OSVOS by updating the network online using instance-level semantic information. PReMVOS [18] integrates different networks with fine-tuning and merging, which achieves superior performance. Online fine-tuning methods achieve good performance, but poor efficiency due to the fine-tuning process at test time.

Recently, some VOS approaches without first-frame fine-tuning have been proposed and achieve very high speed and effectively. One type of these methods is *propagation-based* [32, 35, 2], which usually takes as input the combination of the image and predicted segmentation mask of the previous frame. For instance, RGMP [32] employs a siamese architecture network. One stream encodes the feature of the target frame and the mask of the previous frame while another stream encodes the first frame together with its given ground truth. Another type of fine-tuning free methods is *matching-based* [8, 13, 27, 31], which utilizes the **pixel embedding learning**. For instance, PML [8] learns a pixel embedding space by a triplet loss together with a nearest neighbor classifier. VideoMatch [13] proposes a soft matching mechanism by calculating similarity score maps of matching features to generate smooth predictions. FEELVOS [27] employs pixel-wise embedding together with a global and a local matching mechanism. By considering foreground-background integration, CFBI [36] achieves the new state of the art. Our method is inspired by FEELVOS [27], and utilizes the global and local matching maps to transfer information of the scribble-annotated and previous frame to the target frame.

Interactive Video Object Segmentation. In the interactive VOS setting, users can provide various types of inputs (*e.g.* points, scribbles) to select the objects of interest and refine the segmentation results by providing more interactions. Previous interactive methods [29, 25, 1], either use
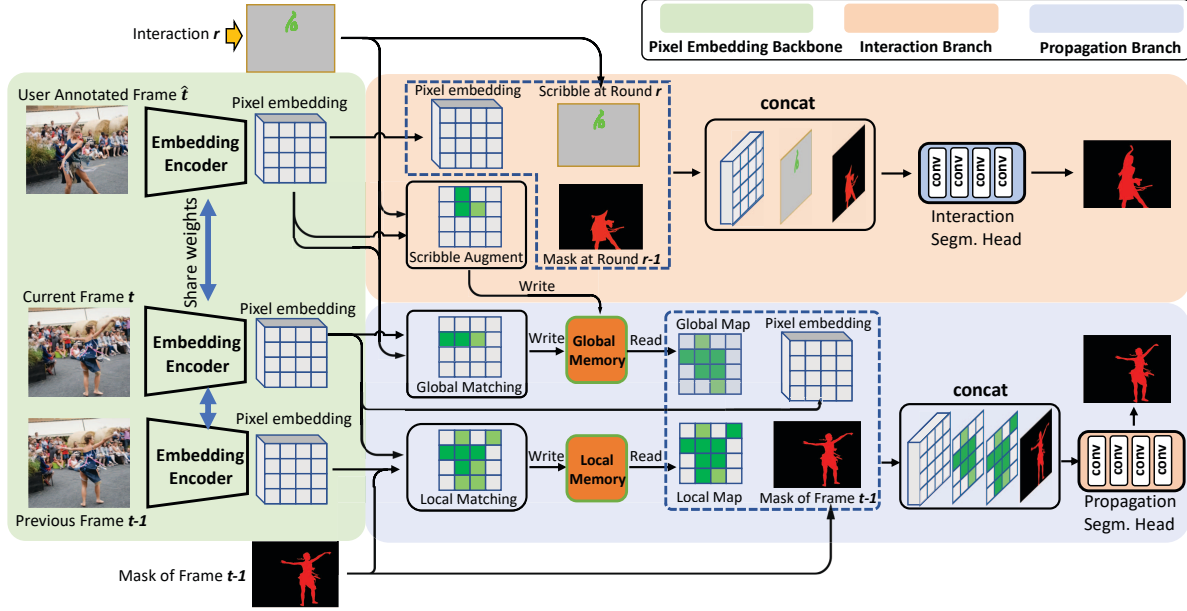
Figure 2. The pipeline of our MA-Net, including the **pixel embedding backbone**, the **interaction branch**, and the **propagation branch**. During inference, the pixel embedding of all frames is extracted only once in the first round. The interaction branch employs "shallow" convolutional layers to predict the mask of the interactive frame. The propagation branch uses a memory aggregation mechanism to record informative knowledge and "shallow" convolutional layers to generate masks of other frames. In the matching processes shown in the figure, the deeper the green, the higher the probability of being predicted as the target object. Best viewed in color.

hand-crafted features or need a lot of interactions, can not achieve a good performance or efficiency.

Recently, some round-based deep learning methods [21, 12, 20, 3] for iVOS have been proposed. Benard *et al*. [3] and Heo [12] treat the interactive VOS as two sub-tasks: using the scribbles to generate segmentation masks, and using the generated mask to infer masks of other frames as semi-supervised VOS. Oh [21] uses two networks, interaction and propagation, to tackle these two sub-tasks. These two networks are connected both internally and externally. These methods [3, 12, 21] have several limitations: (1) they use two independent networks without shared weights, and need new feed-forward computation in each interaction round [21, 12], making it inefficient when rounds grow up; (2) they do not utilize the multi-round information adequately. Recently, Oh [22] proposes a space-time memory mechanism to store informative knowledge and achieves state-of-the-art performance. Different from our memory mechanism, they need complicated key-value computation. Besides, they also need new feed-forward computation in each interaction round, which is time-consuming.

3. Method

Round-based iVOS aims at cutting out the target objects in all frames of a video given user annotations (*e.g.* scribbles) on one frame. Users can provide additional feedback annotations on a frame after reviewing the segmentation results to refine the segmentation mask of the next round.

Previous methods [12, 21, 3] chose to adopt two independent neural networks (interaction and propagation) without shared weights or connect two networks by medial layers, which usually affects the inference efficiency. In this paper, we deal with the two sub-tasks (interaction and propagation) under a unified pixel embedding learning framework.

To this end, we propose MA-Net, which contains three modules: a pixel embedding encoder, an interaction branch, and a propagation branch, as shown in Fig. 2. The pixel embedding encoder takes the RGB frames of the given video as inputs and encodes each pixel into an embedding vector. The interaction branch leverages the user's annotations (scribbles) and the pixel embedding of the user-annotated frame to generate the instance segmentation mask. The propagation branch propagates the informative knowledge of the user-annotated frame and the previous frame to the current frame using the pixel embedding. Both the two branches share weights of the pixel embedding encoder, and then employ two "shallow" networks with several convolutional layers as the segmentation heads, respectively. The pixel embeddings of all frames are extracted only in the first interaction round. During the refinement process in the following rounds, only the two "shallow" segmentation heads are used, making our MA-Net more efficient than previous methods. In this paper, we denote the current processing frame as the tth frame, the previous frame as the $(t-1)^{th}$ frame, and the user-annotated frame as the $\hat{t}^{th}$ frame. Pixels of the current processing frame are denoted as p, and pixels

10365

User Annotated Frame $\hat{t}$ Current Frame t Previous Frame t-1

Pixel Embedding   Pixel Embedding   Pixel Embedding
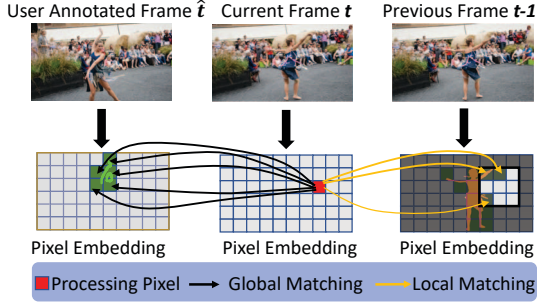
■ Processing Pixel ⟶ Global Matching ⟶ Local Matching

Figure 3. Global matching and local matching process. For each pixel in the current processing frame at time t, distances are calculated with pixels of the target object annotated by scribbles (global map) or predicted mask (local map) and the smallest value of distances (nearest neighbor) are used to construct the matching map.
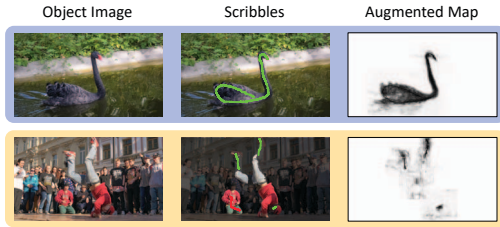

Object Image   Scribbles   Augmented Map

Figure 4. Examples of the augmented map computed by the pixel embedding and scribbles.

annotated or predicted to belong to the target object o as q. In the following, we will describe each of the modules in more detail.

Pixel Embedding Encoder. The purpose of pixel embedding learning is to learn an embedding space where pixels belonging to the same object are close while pixels belonging to different objects are far away. We employ the DeepLabv3+ architecture [7] based on ResNet101 [11] as our backbone, and add an embedding layer consisting of one depth-wise separable convolution with a kernel size of 3×3. The stride of the pixel embedding feature is 4, and the dimension is 100. For each pixel p in the input RGB frame, we learn a semantic embedding vector $\mathbf{e}_p$ in the learned embedding space. In this paper, we encode the pixel embedding into a Euclidean space, where the Euclidean norm between two pixels in the same object is expected to be small. Similar to [10, 27], we define the distance between pixels p and q in terms of their corresponding embedding vectors $\mathbf{e}_p$ and $\mathbf{e}_q$ as

$$d(p,q) = 1 - \frac{2}{1 + \exp(\|\mathbf{e}_p - \mathbf{e}_q\|_2^2)}. \qquad (1)$$

This operation aims at normalizing the pixel distance between 0 and 1. We follow the strategy of FEELVOS [27] to employ the pixel distances as a soft cue, which is further refined by two "shallow" segmentation heads.

Propagation Branch. The propagation branch aims at propagating information from the user-annotated frame and the previous frame to predict the segmentation mask of the target object at the current frame. Following

FEELVOS [27], we employ the global and local matching map as the soft cues of the user-annotated frame and the previous frame, respectively. The matching processes of the global and local maps are shown in Fig. 3. Different from FEELVOS [27], our MA-Net proposes to employ a memory aggregation mechanism to record and aggregate the informative knowledge during the previous multiple interaction rounds, which is specially designed for iVOS.

Global Map Memory. Let $\mathbb{P}_t$ denotes the set of all pixels of the current t^{th} frame and $\mathbb{P}_{\hat{t},o,r}$ denotes the set of user-annotated pixels of the interactive $\hat{t}^{th}$ frame in the rth round of interaction. As show in the left of Fig. 3, for each pixel $p \in \mathbb{P}_t$, we can calculate the distance of its nearest neighbour in $\mathbb{P}_{\hat{t},o,r}$ to construct a global matching distance map, which is defined by

$$\mathbf{G}_{t,r}(p) = \min_{q \in \mathbb{P}_{\hat{t},o,r}} d(p,q). \qquad (2)$$

Different from the semi-supervised VOS who obtains a fully annotated frame, the interactive setting only provides a small number of scribble annotations to the objects of interest in each round. Therefore, the produced global matching map in one round is usually insufficient to discover the entire target object. To tackle this problem, we build a global memory unit to record and aggregate the historical global matching maps to enrich the information of the target object. Let $\mathbf{M}^g \in \mathbb{R}^{n,o,h,w}$ denotes the global map memory, where n, o, h, w denotes the total number of video frames, the target object, the height and width of the embedding feature maps, respectively. Consider that the range of the values in the matching map is from 0 to 1, where the value of pixels closer to 0 is more likely to belong to the selected object and vice versa. We initialize $\mathbf{M}^g$ with 1 and update $\mathbf{M}^g$ by preserving the minimum value of each pixel in different interaction rounds. We demonstrate the updating process of the global map memory in Figure. 5 (a). Formally, for the round of r and the frame at time t, $\mathbf{M}^g$ is *written* by

$$\mathbf{M}^g_{t,r} = \min(\mathbf{M}^g_{t,r-1}, \mathbf{G}_{t,r}). \qquad (3)$$

When we *read* the accumulated global map of round r, we directly use the updated global map memory $\mathbf{M}^g_{t,r}$.

Local Map Memory and Forgetting. Since the motion between two adjacent frames is usually small, to take advantage of the information of predicted mask from the previous frame, we further introduce the local matching map [27]. To avoid false-positive matches as well as save computation time, we only calculate the matching distance map with a small local region. Let $\mathbb{P}_{t-1,o}$ denote the pixels of frame at time $t-1$ which are predicted to be the object o. $\mathbb{N}(p)$ denotes the neighborhood set of pixel p, which contains pixels at most k pixels far away from p. As shown in the right of Fig. 3, for each pixel p belonging to the frame at time t, we
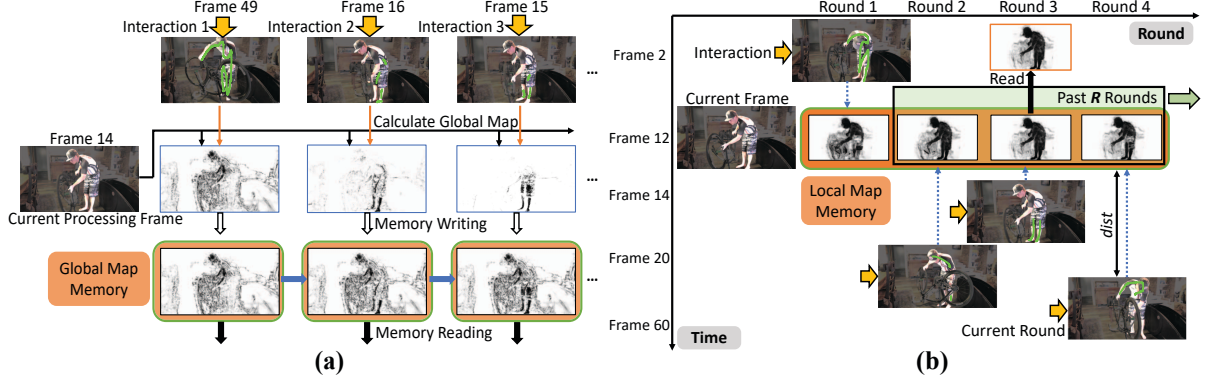
10366

Figure 5. **(a)** The **global map memory mechanism**. The global map in the propagation branch and the augmented map in the interaction branch are recorded and aggregated in the memory. **(b)** The **local map memory and forgetting mechanism**. The local map in each interaction round is recorded in the memory, and the nearest map of time in past R rounds is read to compute the masks. Local maps of early interaction rounds are forgotten with rounds growing up. The blue arrows denote the temporal propagation.

can then compute the local matching distance map $\mathbf{L}_{t,r}$ by

$$\mathbf{L}_{t,r}(p) = \begin{cases} \min_{q \in \mathbb{P}_{t-1,o}^{N}} d(p,q) & \text{if } \mathbb{P}_{t-1,o}^{N} \neq \varnothing \\ 1 & \text{otherwise,} \end{cases} \quad (4)$$

where $\mathbb{P}_{t-1,o}^{N} := \mathbb{P}_{t-1,o} \cap \mathbb{N}(p)$ is the intersection set of the previous frame pixel set $\mathbb{P}_{t-1,o}$ and the neighbour set $\mathbb{N}(p)$.

Different from the scribble annotations provided by users, the mask information of the previous frame is unreliable since the segmentation mask of the previous frame is predicted by the algorithm. In practice, we found that the error will accumulate due to drifts and occlusions during the propagation. The segmentation result will get worse if the current frame *far away* from the user-annotated frame. Therefore, to prevent the error accumulation, we additionally build a local memory unit $\mathbf{M}^{l} \in \mathbb{R}^{n,r,o,h,w}$ to record the historical local matching maps in the previous interaction rounds. Formally, the local map $\mathbf{L}_{t,r}$ for the tth frame in round r is *written* into the local memory by

$$\mathbf{M}_{t,r}^{l} = \mathbf{L}_{t,r}, \quad (5)$$

which means that the writing process of the local memory is simply recording.

When reading from the local memory, for the current t^{th} frame, we calculate the distance of time to the user-annotated $\hat{t}^{th}$ frame of each round r, $dist_{r} = |t - \hat{t}_{r}|$, and select the nearest one to the user-annotated frame as the final local map. However, with the interactive round grows up, the accuracy of segmentation becomes better and better. For instance, a processing frame using the local map of round 8, although far away from the user-annotated frame in this round, may be better than using the local map of round 1 adjacent to the user-annotated frame. Hence we employ a *forgetting mechanism* by using the nearest local map to the user-annotated frame in only past R rounds. Local maps of early interaction rounds will be forgotten. $R = 1$ means we only use local maps of the current round and do not employ

the memory mechanism. The local map memory and forgetting mechanism is shown in Fig. 5 (b). Formally, denote the final local map for the current t^{th} frame in round r^{*} as $\mathbf{L}_{t,r^{*}}'$, then $\mathbf{L}_{t,r^{*}}'$ is *read* from $\mathbf{M}^{l}$ by

$$\mathbf{L}_{t,r^{*}}' = \mathbf{M}_{t,r'}^{l}, r' = \arg \min_{r} |t - \hat{t}_{r}| \; and \; |r' - r^{*}| \leq R \quad (6)$$

We utilize the propagation head with four convolutional layers to predict a one-dimensional map of logits for each selected object. The propagation head takes as input the concatenation of the pixel embedding, the global and local matching map read from memories, and the predicted mask of the previous frame. We stack the logits, apply softmax over the object dimension to obtain the probability map for each pixel.

Interaction Branch. The interaction branch aims at generating a segmentation mask of the user-annotated frame (interactive frame) given user annotations. As shown in Fig. 2, for generating the segmentation mask of the interactive frame in the current round, we concatenate the pixel embedding, the scribbles and the predicted mask from last round along the channel dimension, and use an interaction segmentation head with four convolutional layers to generate the segmentation logits of the target object o. For the multi-object cases, the interaction segmentation head extracts one-dimensional feature maps of logits for all objects, which are then stacked together to obtain the probability map for each pixel by applying the softmax operation over the object dimension.

In iVOS, the interaction branch need not only generate the segmentation mask of the interactive frame in the current round but also record and accumulate informative knowledge of the scribbles for improving the segment results of this frame in the next rounds. We propose a matching map to *augment* the incomplete scribbles by mining the property of the pixel embedding space, and record the augment map into the global memory $\mathbf{M}^{g}$. In the pixel embedding space, the pixels close to the annotated pixels have a
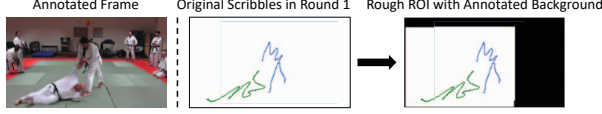
Figure 6. In the first round, there are no annotations of the background. We use a rough ROI and annotate pixels out of ROI as the background (black area). Green and blue scribbles annotate the first and second objects, respectively.

higher probability of belonging to the same object. Similar to the local map proposed in the propagation branch, we employ a matching distance map to augment the scribbles. Suppose $\mathbb{P}_{\hat{t}}$ denote the set of all pixels (with a stride of 4 in the embedding space) of the user-annotated frame at time $\hat{t}$ and $\mathbb{P}_{\hat{t},o}$ denote the set of scribble-annotated pixels belonging to the target object o. For each pixel $p \in \mathbb{P}_{\hat{t}}$, we compute the distance of its nearest neighbor in the annotated pixels $\mathbb{P}_{\hat{t},o}$ to construct the matching distance map. To avoid introducing the unexpected noisy pixels that are similar to the annotated ones but with large spatial distances, for each pixel $p \in \mathbb{P}_{\hat{t}}$, we only consider those pixels within its local neighborhood as the searching candidates. We denote $\mathbb{N}(p)$ as the neighbourhood set of p, where $\mathbb{N}(p)$ contains pixels at most k pixels far away from p. Therefore, the augmented map $\mathbf{A}_{\hat{t}}(p)$ for pixel p is defined by

$$\mathbf{A}_{\hat{t}}(p) = \begin{cases} min_{q \in \mathbb{P}_{\hat{t},o}^{N}} d(p,q) & \text{if } \mathbb{P}_{\hat{t},o}^{N} \neq \varnothing \\ 1 & \text{otherwise,} \end{cases} \quad (7)$$

where $\mathbb{P}_{\hat{t},o}^{N} := \mathbb{P}_{\hat{t},o} \cap \mathbb{N}(p)$ is the intersection set of the scribble-annotated set $\mathbb{P}_{\hat{t},o}$ and the neighbourhood set $\mathbb{N}(p)$. Fig. 4 shows the comparison of the scribbles and the augmented maps, and we can find that the augmented map contains more information about the selected objects. The augmented map $\mathbf{A}_{\hat{t}}$ will be recorded and aggregated in the global map $\mathbf{M}^g$. For the interactive frame at the time $\hat{t}$ in the round of r, $\mathbf{M}^g$ is updated by

$$\mathbf{M}_{\hat{t},r}^g = \min(\mathbf{M}_{\hat{t},r-1}^g, \mathbf{A}_{\hat{t},r}). \quad (8)$$

This operation can benefit the segmentation result of the interactive frame in next rounds.

4. Experiments

4.1. Training and Inference

Training Procedure. We employ a two-stage training procedure to train our MA-Net. In Stage 1, we train the propagation branch with the pixel embedding encoder. To simulate the video propagation process, we randomly select three frames from one training video as a training batch. One of the frames serves as the reference frame, *i.e.*, it plays the role of the frame annotated by scribbles. Two adjacent frames serve as the previous frame and the current processing frame. Some methods [21, 12] leverage the synthesized

scribbles for the reference frame to train the propagation network. However, the synthesized scribbles are all densely generated from the ground-truth masks. After performing the training for a large number of iterations, ground-truth masks are actually used. Since the propagation branch is trained independently and densely generating synthesized scribbles from groundruth in an online manner is often time-consuming, we directly use the ground-truth instance mask of the reference frame. In practice, we found that the reference frame using ground truth achieves similar performance to using the synthesized scribbles during training.

In Stage 2, after training the pixel embedding encoder and the propagation branch, we fixed the pixel embedding encoder and trained the interaction branch. It is not feasible to collect a large number of scribbles annotated by users. Therefore, we train our model with synthesized scribbles. In the first round, we use the scribbles of the training set provided by the DAVIS Challenge 2018 [6]. In the following rounds, scribbles are synthesized within false negative and false positive areas. There is a gap between the first round and the following rounds since the first round only provides positive scribbles while following rounds provide both positive and negative scribbles. Hence we use the background label as the mask of the previous round for the first round.

Inference. We follow the round-based interactive setting of the DAVIS Challenge 2018. In the first round, users provide positive scribbles and no negative scribbles. To eliminate the gap between training and testing, we use a rough Region of Interest (ROI) that contains all positive scribbles and enlarge ROI by enough space to make sure it contains all parts of the target object. Then we annotate all the pixels out of the enlarged ROI as the background (Fig. 6). We extract the pixel embedding of each frame and utilize the interaction branch and propagation branch to generate segmentation masks of the target video. In the following round, users annotate the frame of the video with the worst performance using scribbles. Our model extracts the pixel embeddings of all frames for only once in the first round. The extracted pixel embeddings are further employed to compute the refined segmentation masks with the interaction and propagation heads in the following rounds, leading to our MA-Net more efficient than previous methods.

Implementation Details. We use the DeepLabv3+ architecture [7] based on ResNet101 [11] as our backbone, which produces an output feature maps with a stride of 4. On the top of the backbone, we add an embedding layer consisting of one depth-wise separable convolution with a kernel size of 3×3. The dimension of the pixel embedding is 100 advised by [27].

For the interaction and propagation segmentation heads, we employ four depth-wise separable convolutional layers with a dimension of 256, a kernel size of 7×7 for the depth-wise convolutions, a batch normalization operation and a
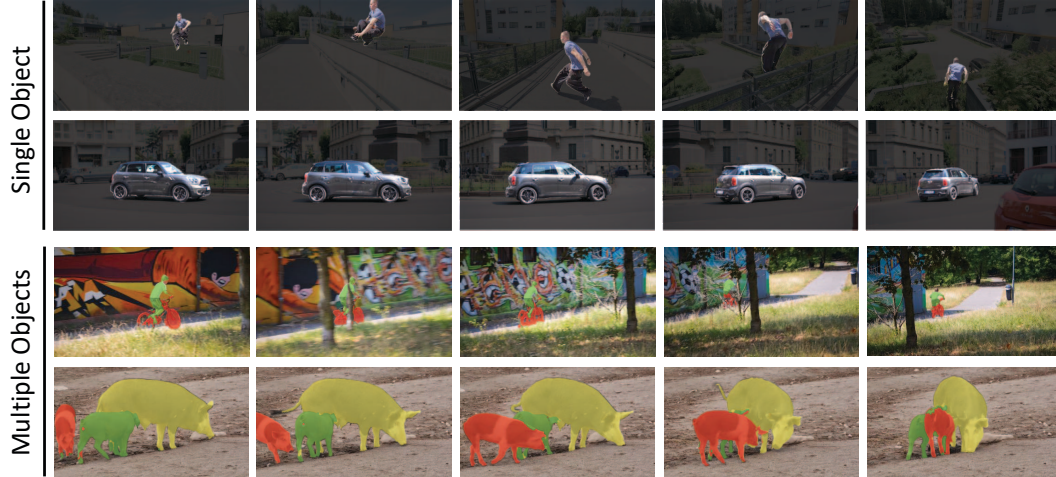
10368

Figure 7. The qualitative results on the DAVIS-2017 validation set. All the user interactions are automatically simulated by the robot agent provided by [6]. All result masks are sampled after 8 rounds.

ReLU activation function. Finally, a 1×1 convolution is employed to extract the predicted logits.

When computing the local matching map, we downsample the pixel embedding by a factor of 2 for computational efficiency. In practice, we set the local window size as $k = 12$ in this paper, considering the trade-off between accuracy and efficiency. We utilize SGD optimization with a learning rate of 0.0007 and a batch size of 2. We employ the adaptive bootstrapped cross-entropy loss [23], which takes into account 100% to 15% hardest pixels from step 0 to step 50000 for computing the loss. All input images are augmented by random flipping, scaling, and cropping. The input size is 416×416 pixels. When processing the training of the first stage, we initialize the weights of the backbone with the weights pre-trained on ImageNet [9] and COCO [17], and we train the pixel embedding encoder and the propagation head on the training set of DAVIS [24] for 100000 steps. When training our model in the second stage, we use a round-based training with three rounds per circle. The first round uses only the positive scribbles while the following two rounds use both the positive and negative scribbles and the previous round masks. We train the second stage on the training set of DAVIS [24] for 80000 steps.

4.2. Results

Evaluating iVOS quantitatively is difficult since the user input is directly related to the segmentation results, and different users may provide different scribbles. To tackle this problem, Caelles et al. [6] proposes a robot agent service to simulate human interaction for a fair comparison.

Quantitative Results. To fairly compare our MA-Net with the state-of-the-art methods, we evaluated our model on the DAVIS validation set following the interactive track benchmark in the DAVIS Challenge 2018 [6]. In this benchmark, a robot agent interacts with each model for 8 rounds,

Method	+OF	+CRF	+YV	AUC	J@60
Najafi et al. [20]		✓		0.702	0.548
Heo et al. [12]			✓	0.698	0.691
Heo et al. [12]	✓		✓	0.704	0.725
Oh et al. [21]			✓	0.691	0.734
MA-Net(Ours)				**0.749**	**0.761**

Table 1. Comparison of our MA-Net with the previous methods on the validation set in DAVIS2017. The entries are ordered according to the J@60 score. +OF denotes using optical flow, +CRF denotes using the CRF [14] as post-processing and +YV denotes using additional YoutubeVOS training set [33] when training.

and the model is expected to compute masks within 30 seconds per interaction for each object. There are two evaluation metrics: area under the curve (AUC) and Jaccard at 60 seconds (J@60s). AUC is designed to measure the overall accuracy of the evaluation. J@60 measures the accuracy with a limited time budget (60 seconds). Table. 1 shows the comparison of our method and previous state-of-the-art iVOS methods. Comparing with the best competing method Heo [12], according to accuracy, our method surpasses it by +4.7% AUC. Comparing with the best competing method Oh et al. [21], according to efficiency, our method surpasses it by +2.7% J@60s. Besides, our model does not use any bells and whistles such as optical flow, post-processing (CRF), or additional video training set, i.e., YoutubeVOS [33]. In addition, our MA-Net can accomplish 7-round interactions within 60seconds, which is more efficient than the state of the art one [21] of 5-round interactions within 60 seconds [1]. In summary, our MA-Net outperforms previous methods in both accuracy and efficiency.

Qualitative Results. Fig. 7 shows qualitative results on the DAVIS 2017 validation set. It can be seen that our MA-Net produces accurate segmentation masks in multiple

---

[1] To fairly compare the efficiency, we test our model on a 1080Ti GPU following Oh [21]

Local window size k	6	9	12	15
AUC	0.724	0.737	0.749	0.748
J@60	0.730	0.753	0.761	0.761

Table 2. The impact of the local window size k.

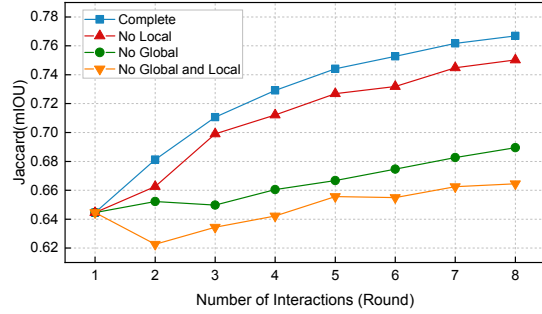Figure 8. Ablation study on DAVIS 2017 validation set to show the effectiveness of our proposed global and local memories.

Figure 9. The impact of R in the local map memory. R denotes that local maps in past R rounds in the memory are used.

cases of large variance, including the single object condition and multiple objects condition. Qualitative results also show that our method can handle the occlusion issue (the 3rd row). In some difficult cases, *e.g.*, a video contains multiple objects of the same class and the objects are occluded by each other (pigs in the 4th row), our method may make mistakes in some similar parts of different objects. This is most likely because the pixel embedding vectors of similar parts are close to each other.

4.3. Ablation Study

The Effectiveness of the Memory Mechanism. We conduct ablation studies using the DAVIS 2017 validation dataset to validate the effectiveness of our proposed memory mechanism. Fig. 8 and Fig. 9 show the Jaccard score of ablation models with growing number of interactions. In Fig. 8, we compare our method with and without global and local memories. **No Global** indicates we use the model without the global memory, which means we only use the global map calculated in the first round and do not aggregate it in the following rounds. **No Local** indicates that we only use the local map calculated in the current round and do not access local maps from previous rounds. **No Global and Local** is a model without using both the global map memory and local map memory. We can find that both the global map memory and the local map memory take effects in the iVOS and greatly improves the performance since utilizing all scribble information of previous rounds.

As described in Section 3, for the memory of the local map, there is a trade-off between choosing the nearest frame and the closest round. In practice, the segmentation mask far away from the annotated frame achieves worse results due to the error accumulation during propagation, so we choose the local map in which round it is nearest to the annotated frame. However, with the interactive round grows up, the accuracy of segmentation becomes better and better. Therefore, we use the nearest map to the annotated
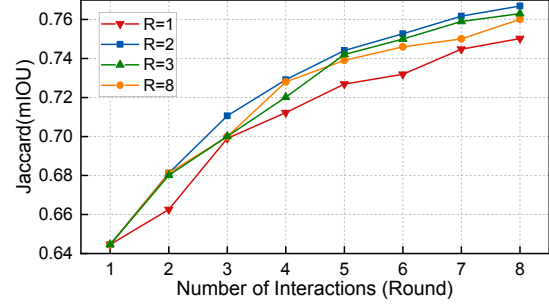
frame in the past R rounds. $R = 1$ means we only use local maps of the current round while $R = 8$ means we use the nearest map in all previous rounds. Fig. 9 shows that when $R > 1$, the segmentation accuracy will improve, indicating the effectiveness of the local map memory. When $R = 2$, our method achieves the best performance, and we choose $R = 2$ for our final model.

The Effectiveness of the Augmented Map. The augmented map of the interactive frame is stored in the global memory in the current interaction round, which will help this frame be correct segmented in the following interaction rounds. Therefore, without the augmented map, the valuable interactive information of this frame will be lost during the propagation in the following interaction rounds. Besides, since our MA-Net also takes local matching into account, the improvements of all the interactive frames will further implicitly bring additional benefits to their subsequent non-interactive frames during the propagation. To be specific, the AUC score will drop from 0.749 to 0.744 if the augment map is removed from the global memory.

The Impact of the Local Window Size. In addition, we also study the impact of the local window size k, as shown in Table. 2. When k is smaller, the local map computation is more efficient. However, a small k will affect the accuracy of our model. In practice, we choose $k = 12$ in this paper.

5. Conclusion

Video object segmentation (VOS) is a fundamental task in computer vision. In this paper, we propose a user-friendly framework to generate accurate segmentation masks of a video with a few user annotations. Our MA-Net integrates the interaction and propagation operations into a unified pixel embedding learning framework, which promotes the efficiency of the round-based interactive VOS. More importantly, we propose a novel memory aggregation mechanism to record and aggregate the information of the user interactions and predictions of previous interaction rounds, which improves the segmentation accuracy greatly.

References

[1] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Transactions on Graphics (ToG)*, 28(3):70, 2009. 1, 2

[2] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, pages 5977–5986, 2018. 2

[3] Arnaud Benard and Michael Gygli. Interactive video object segmentation in the wild. *arXiv preprint arXiv:1801.00269*, 2017. 2, 3

[4] Benjamin Bratt. *Rotoscoping*. Routledge, 2012. 1

[5] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, pages 221–230, 2017. 2

[6] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*, 2018. 1, 2, 6, 7

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 4, 6

[8] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, pages 1189–1198, 2018. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 7

[10] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017. 4

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4, 6

[12] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Interactive video object segmentation using sparse-to-dense networks. *CVPR Workshops*, 2019. 2, 3, 6, 7

[13] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, pages 54–70, 2018. 2

[14] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*, pages 513–521, 2013. 2, 7

[15] Wenbin Li, Fabio Viola, Jonathan Starck, Gabriel J Brostow, and Neill DF Campbell. Roto++: Accelerating professional rotoscoping using shape manifolds. *ACM Transactions on Graphics (TOG)*, 35(4):62, 2016. 1

[16] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, pages 90–105, 2018. 2

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 7

[18] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, pages 565–580. Springer, 2018. 2

[19] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *TPAMI*, 41(6):1515–1530, 2018. 2

[20] Mohammad Najafi, Viveka Kulharia, T Ajanthan, and PH Torr. Similarity learning for dense label transfer. *CVPR Workshops*, 2018. 2, 3, 7

[21] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Fast user-guided video object segmentation by interaction-and-propagation networks. In *CVPR*, pages 5247–5256, 2019. 2, 3, 6, 7

[22] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 2, 3

[23] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *CVPR*, pages 4151–4160, 2017. 7

[24] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 7

[25] Brian L Price, Bryan S Morse, and Scott Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, pages 779–786. IEEE, 2009. 1, 2

[26] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, pages 5277–5286, 2019. 2

[27] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, pages 9481–9490, 2019. 2, 4, 6

[28] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In *CVPR Workshops*, volume 5, 2017. 2

[29] Jue Wang, Pravin Bhat, R Alex Colburn, Maneesh Agrawala, and Michael F Cohen. Interactive video cutout. In *ACM Transactions on Graphics (ToG)*, volume 24, pages 585–594. ACM, 2005. 1, 2

[30] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven CH Hoi, and Haibin Ling. Learning unsupervised video object segmentation through visual attention. In *CVPR*, pages 3064–3074, 2019. 2

[31] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *ICCV*, 2019. 2

[32] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, pages 7376–7385, 2018. 2

[33] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, pages 585–601, 2018. 7

[34] Shuangjie Xu, Daizong Liu, Linchao Bao, Wei Liu, and Pan Zhou. Mhp-vos: Multiple hypotheses propagation for video object segmentation. In *CVPR*, pages 314–323, 2019. 2

[35] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, pages 6499–6507, 2018. 2

[36] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. *arXiv preprint arXiv:2003.08333*, 2020. 2