Kernelized Memory Network for Video Object Segmentation

Hongje Seong, Junhyuk Hyun, and Euntai Kim*

School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
{hjseong,jhhyun,etkim}@yonsei.ac.kr

Abstract. Semi-supervised video object segmentation (VOS) is a task that involves predicting a target object in a video when the ground truth segmentation mask of the target object is given in the first frame. Recently, space-time memory networks (STM) have received significant attention as a promising solution for semi-supervised VOS. However, an important point is overlooked when applying STM to VOS. The solution (STM) is non-local, but the problem (VOS) is predominantly local. To solve the mismatch between STM and VOS, we propose a kernelized memory network (KMN). Before being trained on real videos, our KMN is pre-trained on static images, as in previous works. Unlike in previous works, we use the Hide-and-Seek strategy in pre-training to obtain the best possible results in handling occlusions and segment boundary extraction. The proposed KMN surpasses the state-of-the-art on standard benchmarks by a significant margin (+5% on DAVIS 2017 test-dev set). In addition, the runtime of KMN is 0.12 seconds per frame on the DAVIS 2016 validation set, and the KMN rarely requires extra computation, when compared with STM.

Keywords: Video object segmentation, Memory network, Gaussian kernel, Hide-and-Seek

1 Introduction

Video object segmentation (VOS) is a task that involves tracking target objects at the pixel level in a video. It is one of the most challenging problems in computer vision. VOS can be divided into two categories: semi-supervised VOS and unsupervised VOS. In semi-supervised VOS, the ground truth (GT) segmentation mask is provided in the first frame, and the segmentation mask must be predicted for the subsequent frames. In unsupervised VOS, however, no GT segmentation mask is provided, and the task is to find and segment the salient object in the video. In this paper, we consider semi-supervised VOS.

Space-time memory networks (STM) [30] have recently received significant attention as a promising solution for semi-supervised VOS. The basic idea behind the application of STM to VOS is to use the intermediate frames between the first frame and the current frame. In STM, the current frame is considered to be

* Corresponding author.

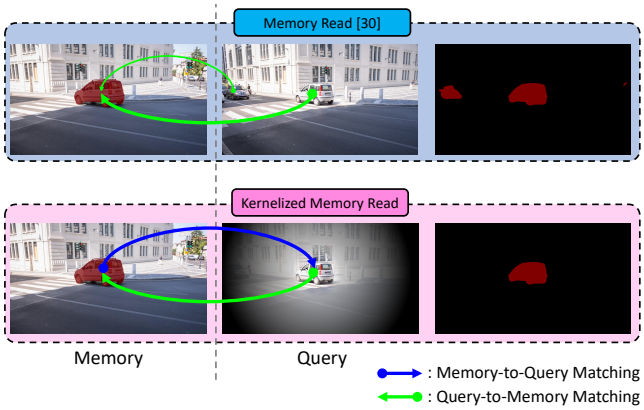

Fig. 1. Illustration of KMN. In STM [30], two cars in the query frame are matched with a car in the memory frame owing to the non-local matching between the query and memory. The car in the middle is the correct match, while the car on the left is an incorrect match. In KMN, however, non-local matching between the query and memory is controlled by the Gaussian kernel. Only the car in the middle of the query frame is matched with the car in the memory.

the query frame for which the target is to be predicted, whereas the past (already predicted) frames are used as memory frames. This approach, however, overlooks an important point. The solution (STM) is non-local, but the problem (VOS) is predominantly local, as illustrated in Fig. 1. Specifically, STM is based on non-local matching between the query frame and memory frames. However, in VOS, the target object in the query frame usually appears in the local neighborhood of the target's appearance in the memory frames. To solve the problem arising from the use of STM for VOS, we propose a kernelized memory network (KMN). In KMN, the Gaussian kernel is employed to reduce the degree of non-localization of the STM and improve the effectiveness of the memory network for VOS.

Before being trained on real videos, our KMN is pre-trained on static images, as in some previous works. In particular, multiple frames based on a random affine transform were used in [29,30]. Unlike the training process in the previous works, however, we employ a Hide-and-Seek strategy during pre-training to obtain the best possible results in handling occlusions and segment boundary extraction. The Hide-and-Seek strategy [38] was initially developed for weakly supervised object localization, but we used it to pre-train the KMN. This provides two key benefits. First, Hide-and-Seek achieves segmentation results that are considerably robust to occlusion. To the best of our knowledge, this is the first time that Hide-and-Seek has been applied to VOS in order to make the predictions robust to occlusion. Second, Hide-and-Seek is used to refine the boundary of the object segment. Because most of the ground truths in segmentation datasets contain unclear and incorrect boundaries, it is fairly challenging to predict accurate boundaries in VOS. The boundaries created by Hide-and-Seek, however, are

clear and accurate. Hide-and-seek appears to provide instructive supervision for clear and precise cuts for objects, as shown in Fig. 4. We conduct experiments on DAVIS 2016, DAVIS 2017, and Youtube-VOS 2018 and significantly outperform all previous methods, even compared with online-learning approaches.

The contributions of this paper can be summarized as follows. First, KMN is developed to reduce the non-locality of the STM and make the memory network more effective for VOS. Second, Hide-and-Seek is used to pre-train the KMN on static images.

2 Related Work

Semi-supervised video object segmentation [33,34,49] is a task involving prediction of the target objects in all frames of a video sequence where information of the target objects is provided in the first frame. Because the object mask for the first frame of the video is given at the test time, many previous studies [37,5,32,2,14,7,1,25,26,44,22,47] fine-tuned their networks on the given mask. This is known as the online-learning strategy. Online-learning methods can provide accurate prediction results, but require considerable time for inference and finding the best hyper-parameters of the model for each sequence. Offline-learning methods [27,16,50,3,43,29,4,52,30] use a fixed parameter set trained on the whole training sequence. Therefore, they can have a fast run time, while achieving comparable accuracy. Our proposed method follows the offline approach.

Memory networks [39] use the query, **key**, and **value** (QKV) concept. The QKV concept is often used when the target information of the current input exists at the other inputs. In this case, memory networks set the current input and the other inputs as the query and memory, respectively. The **key** and **value** are extracted from memory, and the correlation map of the query and memory is generated through a non-local matching operation of the query and **key** feature. Then, the weighted average **value** based on the correlation map is retrieved. The QKV concept is widely used in a variety of tasks, including natural language processing [41,28,20], image processing [31,54], and video recognition [46,10,35]. In VOS, STM [30] has achieved significant success by repurposing the concept of the QKV. However, applications in STM tend to overlook an important feature of VOS, leading to a limitation that will be addressed in this paper.

Kernel soft argmax [21] uses Gaussian kernels on the correlation map to create a gradient propagable argmax function for semantic correspondence. The semantic correspondence task requires only a single matching flow from a source image to a target image for each given source point. However, applying a discrete argmax function on the correlation map makes the network untrainable. To solve this problem, kernel soft argmax applies Gaussian kernels on the correlation map and then averages the correlation scores. Our work is inspired by the kernel soft

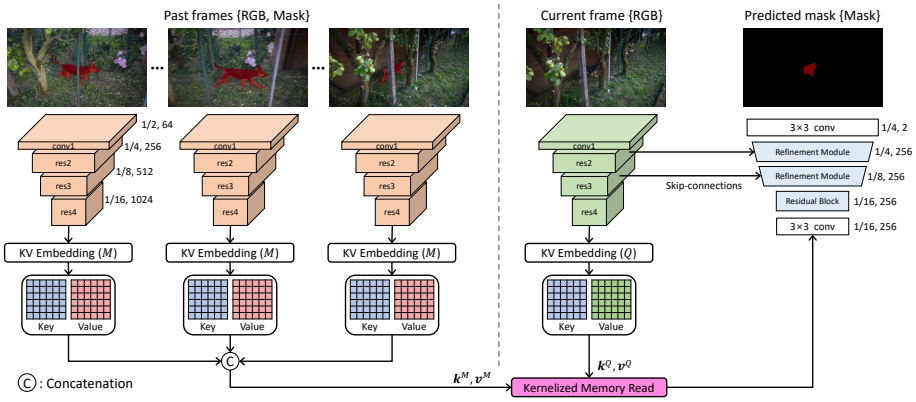

Fig. 2. Overall architecture of our kernelized memory network (KMN). We follow the frameworks of [30] and propose a new operation of kernelized memory read. The numbers next to the block indicate the spatial size and channel dimension, respectively.

argmax, but its application and objective are completely different. The kernel soft argmax applies Gaussian kernels to the results of the searching flow (*i.e.*, memory frame) to serve as a gradient propagable argmax function, whereas we applied Gaussian kernels on the opposite side (*i.e.*, query frame) to solve the case as shown in Fig. 1.

Hide-and-Seek [38] is a weakly supervised framework that has been proposed to improve object localization. Training object localization in a weakly supervised manner using intact images leads to poor localization by finding only the most salient parts of the objects. Hiding some random patches of the object during training helps to improve object localization by forcing the system to find relatively less salient parts. We have found that Hide-and-Seek can improve VOS which is a fully supervised learning task. As a result, we achieved comparable performance to the other offline-learning approaches, even when we trained only on the static images.

Difficulties in segmentation near object boundaries. Although there has been significant progress in image segmentation, accurate segmentation of the object boundary is still challenging. A low-level layer has been trained in EGNet [53] using object boundaries to accurately predict object boundaries. The imbalance between boundary pixels and non-boundary pixels has been addressed in LDF [48] by separating them and training them separately. In this paper, we deal with the problem of GTs that are inaccurate near the object boundary. Hide-and-Seek addresses the problem by generating clean boundaries.

3 Kernelized Memory Network

3.1 Architecture

In this section, we present a kernelized memory network (KMN). The overall architecture of KMN is fairly similar to that of STM [30], as illustrated in Fig. 2. As in STM [30], the current frame is used as the query, while the past frames with the predicted masks are used as the memory. Two ResNet50 [12] are employed to extract the **key** and **value** from the memory and query frames. In memory, the predicted (or given) mask input is concatenated with the RGB channels. Then, the **key** and **value** features of the memory and the query are embedded via a convolutional layer from the `res4` feature [12], which has a 1/16 resolution resolution with respect to the input image. The structures of the **key** and **value** embedding layers for the query and memory are the same, but the weights are not shared. The memory may take several frames, and all frames in the memory are independently embedded and then concatenated along the temporal dimension. In the query, because it takes a single frame, the embedded **key** and **value** are directly used for memory reading.

The correlation map between the query and memory is generated by applying the inner product to all possible combinations of **key** features in the query and memory. From the correlation map, highly matched pixels are retrieved through a *kernelized memory read* operation, and the corresponding **values** of the matched pixels in the memory are concatenated with the **value** of the query. Subsequently, the concatenated value tensor is fed to a decoder consisting of a residual block [13] and two stacks of refinement modules. The refinement module is the same as that used in [30,29]. We recommend that the readers refer to [30] for more details about the decoder.

The main innovation in KMN, distinct from STM [30], lies in the memory read operation. In the memory read of STM [30], only `Query-to-Memory` matching is conducted. In the kernelized memory read of KMN. however, both `Query-to-Memory` matching and `Memory-to-Query` matching are conducted. A detailed explanation of the kernelized memory read is provided in the next subsection.

3.2 Kernelized Memory Read

In the memory read operation of STM [30], the non-local correlation map $c(\mathbf{p}, \mathbf{q})$ is generated using the embedded **key** of the memory $\mathbf{k}^M = \left\{k^M(\mathbf{p})\right\} \in \mathbb{R}^{T\times H\times W\times C/8}$ and query $\mathbf{k}^Q = \left\{k^Q(\mathbf{q})\right\} \in \mathbb{R}^{H\times W\times C/8}$ as follows:

$$c\left(\mathbf{p}, \mathbf{q}\right) = k^M(\mathbf{p})k^Q(\mathbf{q})^\top \tag{1}$$

where H, W, and C are the height, width, and channel size of `res4` [12], respectively. $\mathbf{p} = [p_t, p_y, p_x]$ and $\mathbf{q} = [q_y, q_x]$ indicate the grid cell positions of the **key** features. Then, the query at position $\mathbf{q}$ retrieves the corresponding **value** from

the memory using the correlation map by

$$r(\mathbf{q}) = \sum_{\mathbf{p}} \frac{\exp(c(\mathbf{p},\mathbf{q}))}{\sum_{\mathbf{p}} \exp(c(\mathbf{p},\mathbf{q}))} v^M(\mathbf{p}) \tag{2}$$

where $\mathbf{v}^M = \{v^M(\mathbf{p})\} \in \mathbb{R}^{T\times H\times W\times C/2}$ is the embedded **value** of the memory. Then the retrieved **value** $r(\mathbf{q})$, which is of size $H \times W \times C/2$, is concatenated with the query **value** $\mathbf{v}^Q \in \mathbb{R}^{H\times W\times C/2}$, and the concatenation result is fed to the decoder.

The memory read operation of STM [30] has two inherent problems. First, every grid in the query frame searches the memory frames for a target object, but not vice versa. That is, there is only `Query-to-Memory` matching in the STM. Thus, when multiple objects in the query frame look like a target object, all of them can be matched with the same target object in the memory frames. Second, the non-local matching in the STM can be ineffective in VOS, because it overlooks the fact that the target object in the query should appear where it previously was in the memory frames.

To solve these problems, we propose a kernelized memory read operation using 2D Gaussian kernels. First, the non-local correlation map $c(\mathbf{p},\mathbf{q}) = k^M(\mathbf{p})k^Q(\mathbf{q})^\top$ between the query and memory is computed as in STM. Second, for each grid $\mathbf{p}$ in the memory frames, the best-matched query position $\widehat{\mathbf{q}}(\mathbf{p}) = [\widehat{q}_y(\mathbf{p}), \widehat{q}_x(\mathbf{p})]$ is searched by

$$\widehat{\mathbf{q}}(\mathbf{p}) = \arg\max_{\mathbf{q}} c(\mathbf{p},\mathbf{q}). \tag{3}$$

This is a `Memory-to-Query` matching. Third, a 2D Gaussian kernel $\mathbf{g} = \{g(\mathbf{p},\mathbf{q})\} \in \mathbb{R}^{T\times H\times W\times H\times W}$ centered on $\widehat{\mathbf{q}}(\mathbf{p})$ is computed by

$$g(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{(q_y - \widehat{q}_y(\mathbf{p}))^2 + (q_x - \widehat{q}_x(\mathbf{p}))^2}{2\sigma^2}\right) \tag{4}$$

where σ is the standard deviation. Using Gaussian kernels, the **value** in the memory is retrieved in a local manner as follows:

$$r^k(\mathbf{q}) = \sum_{\mathbf{p}} \frac{\exp\left(c(\mathbf{p},\mathbf{q})/\sqrt{d}\right) g(\mathbf{p},\mathbf{q})}{\sum_{\mathbf{p}} \exp\left(c(\mathbf{p},\mathbf{q})/\sqrt{d}\right) g(\mathbf{p},\mathbf{q})} v^M(\mathbf{p}) \tag{5}$$

where d is the channel size of the **key**. This is a `Query-to-Memory` matching. Here, $\frac{1}{\sqrt{d}}$ is a scaling factor adopted from [41], to prevent the argument in the softmax from becoming large in magnitude, or equivalently, to prevent the softmax from becoming saturated. The kernelized memory read operation is summarized in Fig. 3.

4 Pre-training by Hide-and-Seek

As in previous studies [32,29,30], our KMN is pre-trained using static image datasets that include foreground object masks [9,24,11,36,6,45]. The basic idea of

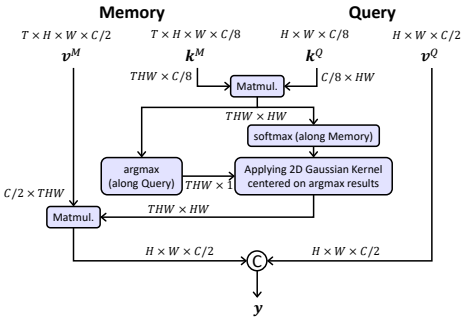

Fig. 3. Kernelized memory read operation.

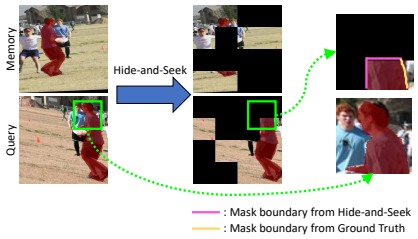

Fig. 4. A pair of images generated during pre-training using Hide-and-Seek. The mask indicated in red denotes the ground truth of the target object.

pre-training a VOS network is to synthetically generate a video with foreground object masks from a single static image. Applying random affine transforms to a static image and the corresponding object mask can yield a synthetic video, and the video can be used to pre-train a VOS network. The problem with synthetic generation of a video from a static image, however, is that the occlusion of the target object does not occur in a generated video. Thus, the simulated video cannot train the pre-trained KMN to cope with the common problem of occlusion in VOS. To solve this problem, the Hide-and-Seek strategy is used to synthetically generate a video with occlusions. Some patches are randomly hidden or blocked, and the occlusions are synthetically generated in the training samples. Here, we only consider squared occluders, but any shape can be taken. Hide-and-Seek can pre-train KMN to be robust to occlusion in the VOS. This idea is illustrated in Fig. 4.

Further, it should be noted that most segmentation datasets contain inaccurate masks (GTs) near the object boundaries. Pre-training KMN with accurate masks is of great importance for high-performance VOS, because inaccurate masks can lead to performance degradation. Manual correction of incorrect masks would be helpful, but it would require a tremendous amount of labor. Another benefit obtained by the use of Hide-and-Seek in pre-training KMN is that the boundaries of the object segment become cleaner and more accurate than before. An example is illustrated in Fig. 4. In this figure, the ground truth mask contains incorrect boundaries on the head of the running person. However, Hide-and-Seek creates a clear object boundary, as represented by the pink line in Fig. 4. A detailed experimental analysis is given in Section 5.6.

The use of Hide-and-Seek in the pre-training on simulated videos significantly improves the VOS pre-training performance; the results are given in Table 1. The pre-training performance obtained by Hide-and-Seek is much higher than that of the previous methods [29,30], and the performance is even as high as the full-training performance of some previous methods.

5 Experiments

In this section, we describe the implementation details of the method, our experimental results on DAVIS 2016, DAVIS 2017, and Youtube-VOS 2018, and the analysis of our proposed methods.

5.1 Training Details

We divide the training stage into two phases: one for pre-training on the static images and another for the main training on VOS datasets composed of video sequences.

During the pre-training, we generated three frames using a single static image by randomly applying rotation, flip, color jittering, and cropping, similar to [29,30]. We then used the Hide-and-Seek framework, as described in Section 4. We first divided the image into a 24×24 grid, which has the same spatial size as the **key** feature. Each cell in the grid had a uniform probability to be hidden. We gradually increased the probability from 0 to 0.5.

During the main training, we followed the STM training strategy [30]. We sampled the three frames from a single video. They were sampled in time-order with intervals randomly selected in the range of the maximum interval. In the training process, the maximum interval is gradually increased from 0 to 25.

For both training phases, we used the dynamic memory strategy [30]. To deal with multi-object segmentation, a soft aggregation operation [30] was used. Note that the Gaussian kernel was not applied during training. Because the argmax function, which determines the center point of the Gaussian kernel, is a discrete function, the error of the argmax cannot be propagated backward during training. Thus, if the Gaussian kernel is used during training, it attempts to optimize networks based on the incorrectly selected **key** feature by argmax, which leads to performance degradation.

Other training details are as follows: randomly resize and crop the images to the size of 384×384, use the mini-batch size of 4, minimize the cross-entropy loss for every pixel-level prediction, and opt for Adam optimizer [19] with a fixed learning rate of 1e-5.

5.2 Inference Details

Our network utilizes intermediate frames to obtain rich information about the target objects. For the inputs of the memory, intermediate frames use the softmax output of the network directly, while the first frame uses the given ground truth mask. Even though we predict all the frames in a sequence, using all the past frames as memory is not only computationally inefficient but also requires considerable GPU memory. Therefore, we follow the memory management strategy described in [30]. Both the first and previous frames are always used. The other intermediate frames are selected at five-frame intervals. Remainders are dropped.

Table 1. Comparisons on the DAVIS 2016 and DAVIS 2017 validation set where ground truths are available. 'OL' indicates the use of online-learning strategy. The best results are **bold-faced**, and the second best results are underlined.

Training Data	Methods	OL	DAVIS 2016 val Time	$\mathcal{G}_\mathcal{M}$	$\mathcal{J}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	DAVIS 2017 val $\mathcal{G}_\mathcal{M}$	$\mathcal{J}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$
Static Images	RGMP [29]		0.13s	57.1	55.0	59.1	-	-	-
	STM [30]		0.16s	-	-	-	60.0	57.9	62.1
	KMN (ours)		0.12s	**74.8**	**74.7**	**74.8**	**68.9**	**67.1**	**70.8**
DAVIS	BVS [27]		0.37s	59.4	60.0	58.8	-	-	-
	OSMN [50]		-	-	-	-	54.8	52.5	57.1
	OFL [40]		120s	65.7	68.0	63.4	-	-	-
	PLM [37]	✓	0.3s	66.0	70.0	62.0	-	-	-
	VPN [16]		0.63s	67.9	70.2	65.5	-	-	-
	OSMN [50]		0.14s	73.5	74.0	72.9	-	-	-
	SFL [5]	✓	7.9s	74.7	74.8	74.5	-	-	-
	PML [3]		0.27s	77.4	75.5	79.3	-	-	-
	MSK [32]	✓	12s	77.6	79.7	75.4	-	-	-
	OSVOS [2]	✓	9s	80.2	79.8	80.6	60.3	56.6	63.9
	MaskRNN [14]	✓	-	80.8	80.7	80.9	-	60.5	-
	VidMatch [15]		0.32s	-	81.0	-	62.4	56.5	68.2
	FAVOS [4]		1.8s	81.0	82.4	79.5	58.2	54.6	61.8
	LSE [7]	✓	-	81.6	82.9	80.3	-	-	-
	FEELVOS [43]		0.45s	81.7	80.3	83.1	69.1	65.9	72.3
	RGMP [29]		0.13s	81.8	81.5	82.0	66.7	64.8	68.6
	DTN [52]		0.07s	83.6	83.7	83.5	-	-	-
	CINN [1]	✓	>30s	84.2	83.4	85.0	70.7	67.2	74.2
	DyeNet [22]		0.42s	-	84.7	-	69.1	67.3	71.0
	RaNet [47]		0.03s	85.5	85.5	85.4	65.7	63.2	68.2
	AGSS-VOS [23]		-	-	-	-	66.6	63.4	69.8
	DTN [52]		-	-	-	-	67.4	64.2	70.6
	OnAVOS [44]	✓	13s	85.5	86.1	84.9	67.9	64.5	71.2
	OSVOS^S [26]	✓	4.5s	86.0	85.6	86.4	68.0	64.7	71.3
	DMM-Net [51]		-	-	-	-	70.7	68.1	73.3
	STM [30]		0.16s	86.5	84.8	88.1	71.6	69.2	74.0
	PReMVOS [25]	✓	32.8s	86.8	84.9	**88.6**	**77.8**	73.9	**81.7**
	DyeNet [22]	✓	2.32s	-	86.2	-	-	-	-
	RaNet [47]	✓	4s	87.1	86.6	87.6	-	-	-
	KMN (ours)		0.12s	**87.6**	**87.1**	88.1	76.0	**74.2**	77.8
+Youtube-VOS	S2S [49]	✓	9s	-	79.1	-	-	-	-
	AGSS-VOS [23]		-	-	-	-	67.4	64.9	69.9
	A-GAME [17]		0.07s	-	82.0	-	70.0	67.2	72.7
	FEELVOS [43]		0.45s	81.7	81.1	82.2	72.0	69.1	74.0
	STM [30]		0.16s	89.3	88.7	89.9	81.8	79.2	84.3
	KMN (ours)		0.12s	**90.5**	**89.5**	**91.5**	**82.8**	**80.0**	**85.6**

Table 2. Comparisons on the DAVIS 2017 test-dev and Youtube-VOS 2018 validation sets where ground truths are unavailable. 'OL' indicates the use of online-learning strategy. The best results are **bold-faced**, and the second best results are underlined.

Methods	OL	DAVIS17 test-dev			Youtube-VOS 2018 val				
		$\mathcal{G}_\mathcal{M}$	$\mathcal{J}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	Overall	$\mathcal{J}_\mathcal{S}$	$\mathcal{J}_\mathcal{U}$	$\mathcal{F}_\mathcal{S}$	$\mathcal{F}_\mathcal{U}$
OSMN [50]		39.3	33.7	44.9	51.2	60.0	40.6	60.1	44.0
FAVOS [4]		43.6	42.9	44.2	-	-	-	-	-
DMM-Net+ [51]		-	-	-	51.7	58.3	41.6	60.7	46.3
MSK [32]	✓	-	-	-	53.1	59.9	45.0	59.5	47.9
OSVOS [2]	✓	50.9	47.0	54.8	58.8	59.8	54.2	60.5	60.7
CapsuleVOS [8]		51.3	47.4	55.2	62.3	67.3	53.7	68.1	59.9
OnAVOS [44]	✓	52.8	49.9	55.7	55.2	60.1	46.6	62.7	51.4
RGMP [29]		52.9	51.3	54.4	53.8	59.5	45.2	-	-
RaNet [47]		53.4	55.3	57.2	-	-	-	-	-
OSVOS^S [26]	✓	57.5	52.9	62.1	-	-	-	-	-
FEELVOS [43]		57.8	55.1	60.4	-	-	-	-	-
RVOS [42]		-	-	-	56.8	63.6	45.5	67.2	51.0
DMM-Net+ [51]	✓	-	-	-	58.0	60.3	50.6	53.5	57.4
S2S [49]	✓	-	-	-	64.4	71.0	55.5	70.0	61.2
A-GAME [17]		-	-	-	66.1	67.8	60.8	-	-
AGSS-VOS [23]		-	-	-	71.3	71.3	65.5	75.2	73.1
Lucid [18]	✓	66.7	63.4	69.9	-	-	-	-	-
CINN [1]	✓	67.5	64.5	70.5	-	-	-	-	-
DyeNet [22]	✓	68.2	65.8	70.5	-	-	-	-	-
PReMVOS [25]	✓	71.6	67.5	75.7	-	-	-	-	-
STM [30]		72.2	69.3	75.2	79.4	79.7	72.8	84.2	80.9
KMN (ours)		**77.2**	**74.1**	**80.3**	**81.4**	**81.4**	**75.3**	**85.6**	**83.3**

We empirically set the fixed standard deviation σ of the Gaussian kernel in (4) to 7. We did not utilize any test time augmentation (*e.g.*, multi-crop testing) or post-processing (*e.g.*, CRF) and used the original image without any pre-processing (*e.g.*, optical flow).

5.3 DAVIS 2016 and 2017

DAVIS 2016 [33] is an object-level annotated dataset that contains 20 video sequences with a single target per video for validation. DAVIS 2017 [34] is an instance-level annotated dataset that contains 30 video sequences with multiple targets per video for validation. Both DAVIS validation sets are most commonly used in VOS to validate proposed methods. We measure the official metrics: the mean of the region similarity $\mathcal{J}_\mathcal{M}$, the contour accuracy $\mathcal{F}_\mathcal{M}$, and their average value $\mathcal{G}_\mathcal{M}$. We used a single parameter set that was trained on the DAVIS 2017 training dataset, which contains 60 video sequences, to evaluate our model on DAVIS 2016 and DAVIS 2017 for a fair comparison with previous works [29,50,30]. The experimental results on the DAVIS 2016 and 2017 validation sets are given in Table 1. We report three different results for each training data.

Fig. 5. Qualitative results and comparisons on the DAVIS 2017 validation set. Our results also do not utilize additional training set, Youtube-VOS.

The results of the training with only static images show a significant margin of improvement from previous studies. In addition, the performances of our proposed network trained on the static images show results comparable to those of the other approaches trained on DAVIS. This indicates that our Hide-and-Seek pre-training approach uses the static images effectively for VOS in training. STM [30] trained on DAVIS showed weak performance compared with the online-learning methods. However, our approach achieves almost similar or even higher performance than the online-learning methods, along with a fast runtime. Finally, the results trained on an additional training dataset, Youtube-VOS, showed the best performance among all existing VOS approaches. Because the ground truths of the DAVIS validation set are accessible to every user, tuning on the dataset is relatively easy. Therefore, to show that a method actually works well in general, we evaluate our approaches on the DAVIS 2017 test-dev benchmark, where ground truths are unavailable, with results shown in Table 2. In DAVIS 2017 test-dev experiments, for a fair comparison, we resize the input frame to be 600p as in STM [30]. We find that our approach surpasses the state-of-the-art method by a significant margin (+5% $\mathcal{G}_{\mathcal{M}}$ score).

5.4 Youtube-VOS 2018

Youtube-VOS 2018 [49] is the largest video object segmentation dataset. It contains 4,453 video sequences with multiple targets per video. To validate on Youtube-VOS 2018, both metrics $\mathcal{J}$ and $\mathcal{F}$ were calculated separately, depending on whether the object categories are seen or not during training: seen sequences with the number of 65 for $\mathcal{J}_{\mathcal{S}}$, $\mathcal{F}_{\mathcal{S}}$, and unseen sequences with the number of 26 for $\mathcal{J}_{\mathcal{U}}$, $\mathcal{F}_{\mathcal{U}}$. The ground truths of the Youtube-VOS 2018 validation set are unavailable as the DAVIS 2017 test-dev benchmark. As shown in Table 2, our approach achieved state-of-the-art performance. This indicates that our approach works well in all cases.

Table 3. Ablation study of our proposed methods. 'HaS' and 'KM' indicate the use of Hide-and-Seek pre-training and kernelized memory read operation, respectively. Note that we did not use additional VOS training data for the ablation study. Only either DAVIS or Youtube-VOS is used, depending on the target evaluation benchmark.

			DAVIS16				DAVIS17			Youtube-VOS 2018				
	HaS	KM	Time*	$\mathcal{G}_\mathcal{M}$	$\mathcal{J}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{G}_\mathcal{M}$	$\mathcal{J}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	Overall	$\mathcal{J}_\mathcal{S}$	$\mathcal{J}_\mathcal{U}$	$\mathcal{F}_\mathcal{S}$	$\mathcal{F}_\mathcal{U}$
STM[30]			0.11s	86.5	84.8	**88.1**	71.6	69.2	74.0	79.4	79.7	72.8	84.2	80.9
Ours			0.11s	81.3	80.0	82.6	72.6	70.1	75.0	79.0	79.2	73.5	83.1	80.3
	✓		0.11s	87.1	86.3	88.0	75.9	73.7	**78.1**	79.5	80.0	73.1	83.9	81.0
		✓	0.12s	87.2	86.6	87.7	73.5	71.2	75.7	81.0	81.0	**75.4**	85.0	82.5
	✓	✓	0.12s	**87.6**	**87.1**	**88.1**	**76.0**	**74.2**	77.8	**81.4**	**81.4**	75.3	**85.6**	**83.3**

* measured on our 1080Ti GPU system

Fig. 6. Qualitative results and comparisons with STM [30]. The noticeable improvements are marked with yellow boxes. For DAVIS results, Youtube-VOS is additionally used for training. Note that the ground truths of the Youtube-VOS validation set are not available.

5.5 Qualitative Results

A qualitative comparison is shown in Fig. 5. We compare our method with the state-of-the-art methods officially released on DAVIS[1]. The other methods in the figure do not utilize any additional VOS training data. Therefore, we show the KMN results which trained only on DAVIS in the main training stage for a fair comparison. Our results show consistently accurate predictions compared to other methods, even in cases of fast deformation (dance-twirl), the appearance of other objects, which are regarded as a background similar to the target object (car-roundabout), and the severe occlusion of the target objects (India).

5.6 Analysis

Ablation study. We conducted an ablation study to demonstrate the effectiveness of our approaches, and the experimental results are presented in Table 3. As shown in the table, our approaches lead to performance improvements. The runtimes were measured on our 1080Ti GPU system, which is the same as that used in [30].

Qualitative comparison with STM. We conducted a qualitative comparison with STM [30], and the results are shown in Fig. 6. To show the actual improvements from STM, we obtained STM results using the author's officially released source code[2]. However, since the parameters for Youtube-VOS validation are not available, our parameters shown in Table 3 were used for Youtube-VOS. For DAVIS, additional data, the Youtube-VOS set, was used for training. As shown in Fig. 6, our results are robust and accurate even in difficult cases where *multiple similar objects appear in the query* and *occlusion occurs*.

Boundary quality made by Hide-and-Seek. To verify that Hide-and-Seek modified the ground truth boundary accurately, we visualized the prediction loss for each pixel in Fig. 7. For a fair comparison, a single model trained on static images was used. As shown in the figure, *most of the losses occur near the boundary*, even when the network predicts quite accurately. This indicates that the networks struggle to learn the mask boundary because the ground truth mask has an irregular and noisy boundary. However, *the boundary of the hidden patch is not activated* in the figure. This means that the network can learn the mask boundary modified by Hide-and-Seek. Thus, Hide-and-Seek can provide more precise boundaries, and we expect that our new perspective would provide an opportunity to improve not only the quality of the segmentation masks, but also system performance for various segmentation tasks in the computer vision field.

[1] https://davischallenge.org/davis2017/soa_compare.html
[2] https://github.com/seoungwugoh/STM

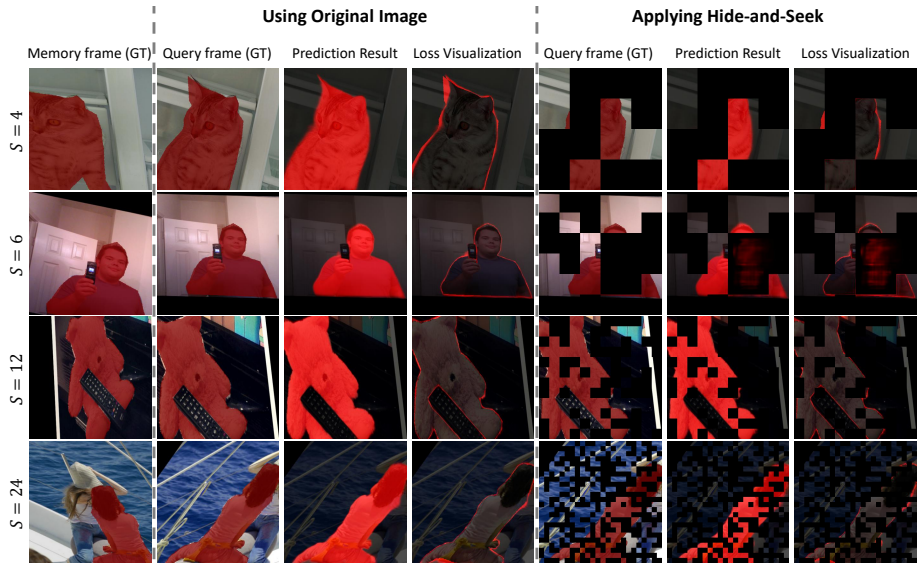

Fig. 7. Pixel-level cross-entropy loss visualization during the pre-training on static images. 'S' indicates the gird size of the Hide-and-Seek. Even if the network finds the object accurately, pixel-level losses occur near the mask boundary, because the ground truth masks near the boundary are not accurate. This makes it difficult for the network to learn the boundary correctly. Since Hide-and-Seek can cut the object cleanly, it gives a more accurate ground truth mask near the boundary. Therefore, we can observe that the losses are not activated on the boundaries made by Hide-and-Seek.

6 Conclusion

In this work, we present a new memory read operation and a method for handling occlusion and obtaining an accurate boundary using a static image. Our proposed methods were evaluated on the DAVIS 2016, DAVIS 2017, and Youtube-VOS benchmarks. We achieved state-of-the-art performance, even including online-learning methods. The ablation study shows the efficacy of our kernel approach, which addresses the main problem of memory networks in VOS. New approaches using the Hide-and-Seek strategy also show its effectiveness for VOS. Since our approaches can be easily reproduced and lead to significant improvements, we believe that our ideas have the potential to improve not only VOS, but also other segmentation-related fields.

Acknowledgement.

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7069370).

References

1. Bao, L., Wu, B., Liu, W.: Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In: CVPR. pp. 5977–5986 (2018) 3, 9, 10
2. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR. pp. 221–230 (2017) 3, 9, 10
3. Chen, Y., Pont-Tuset, J., Montes, A., Van Gool, L.: Blazingly fast video object segmentation with pixel-wise metric learning. In: CVPR. pp. 1189–1198 (2018) 3, 9
4. Cheng, J., Tsai, Y.H., Hung, W.C., Wang, S., Yang, M.H.: Fast and accurate online video object segmentation via tracking parts. In: CVPR. pp. 7415–7424 (2018) 3, 9, 10
5. Cheng, J., Tsai, Y.H., Wang, S., Yang, M.H.: Segflow: Joint learning for video object segmentation and optical flow. In: ICCV. pp. 686–695 (2017) 3, 9
6. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.M.: Global contrast based salient region detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3), 569–582 (2014) 6
7. Ci, H., Wang, C., Wang, Y.: Video object segmentation by learning location-sensitive embeddings. In: ECCV. pp. 501–516 (2018) 3, 9
8. Duarte, K., Rawat, Y.S., Shah, M.: Capsulevos: Semi-supervised video object segmentation using capsule routing. In: ICCV (October 2019) 10
9. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (2010) 6
10. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: CVPR. pp. 244–253 (2019) 3
11. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV. pp. 991–998. IEEE (2011) 6
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 5
13. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV. pp. 630–645. Springer (2016) 5
14. Hu, Y.T., Huang, J.B., Schwing, A.: Maskrnn: Instance level video object segmentation. In: NIPS. pp. 325–334 (2017) 3, 9
15. Hu, Y.T., Huang, J.B., Schwing, A.G.: Videomatch: Matching based video object segmentation. In: ECCV. pp. 54–70 (2018) 9
16. Jampani, V., Gadde, R., Gehler, P.V.: Video propagation networks. In: CVPR. pp. 451–461 (2017) 3, 9
17. Johnander, J., Danelljan, M., Brissman, E., Khan, F.S., Felsberg, M.: A generative appearance model for end-to-end video object segmentation. In: CVPR. pp. 8953–8962 (2019) 9, 10
18. Khoreva, A., Benenson, R., Ilg, E., Brox, T., Schiele, B.: Lucid data dreaming for video object segmentation. International Journal of Computer Vision **127**(9), 1175–1197 (2019) 10
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 8
20. Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., Socher, R.: Ask me anything: Dynamic memory networks for natural language processing. In: ICML. pp. 1378–1387 (2016) 3

21. Lee, J., Kim, D., Ponce, J., Ham, B.: Sfnet: Learning object-aware semantic correspondence. In: CVPR. pp. 2278–2287 (2019) 3
22. Li, X., Change Loy, C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: ECCV. pp. 90–105 (2018) 3, 9, 10
23. Lin, H., Qi, X., Jia, J.: Agss-vos: Attention guided single-shot video object segmentation. In: ICCV (October 2019) 9, 10
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014) 6
25. Luiten, J., Voigtlaender, P., Leibe, B.: Premvos: Proposal-generation, refinement and merging for video object segmentation. In: ACCV. pp. 565–580. Springer (2018) 3, 9, 10
26. Maninis, K.K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: Video object segmentation without temporal information. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(6), 1515–1530 (2018) 3, 9, 10
27. Märki, N., Perazzi, F., Wang, O., Sorkine-Hornung, A.: Bilateral space video segmentation. In: CVPR. pp. 743–751 (2016) 3, 9
28. Miller, A., Fisch, A., Dodge, J., Karimi, A.H., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. In: EMNLP (2016) 3
29. Oh, S.W., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: CVPR. pp. 7376–7385 (2018) 2, 3, 5, 6, 7, 8, 9, 10
30. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (October 2019) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
31. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image transformer. In: ICML. pp. 4052–4061 (2018) 3
32. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR. pp. 2663–2672 (2017) 3, 6, 9, 10
33. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR. pp. 724–732 (2016) 3, 10
34. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017) 3, 10
35. Seong, H., Hyun, J., Kim, E.: Video multitask transformer network. In: ICCV Workshop (2019) 3
36. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended cssd. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(4), 717–729 (2015) 6
37. Shin Yoon, J., Rameau, F., Kim, J., Lee, S., Shin, S., So Kweon, I.: Pixel-level matching for video object segmentation using convolutional neural networks. In: CVPR. pp. 2167–2176 (2017) 3, 9
38. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: ICCV. pp. 3544–3553. IEEE (2017) 2, 4
39. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: NIPS. pp. 2440–2448 (2015) 3

40. Tsai, Y.H., Yang, M.H., Black, M.J.: Video segmentation via object flow. In: CVPR. pp. 3899–3908 (2016) 9
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS. pp. 5998–6008 (2017) 3, 6
42. Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., Giro-i Nieto, X.: Rvos: End-to-end recurrent network for video object segmentation. In: CVPR. pp. 5277–5286 (2019) 10
43. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: Feelvos: Fast end-to-end embedding learning for video object segmentation. In: CVPR. pp. 9481–9490 (2019) 3, 9, 10
44. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017) 3, 9, 10
45. Wang, J., Jiang, H., Yuan, Z., Cheng, M.M., Hu, X., Zheng, N.: Salient object detection: A discriminative regional feature integration approach. International Journal of Computer Vision **123**(2), 251–268 (2017) 6
46. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR. pp. 7794–7803 (2018) 3
47. Wang, Z., Xu, J., Liu, L., Zhu, F., Shao, L.: Ranet: Ranking attention network for fast video object segmentation. In: ICCV (October 2019) 3, 9, 10
48. Wei, J., Wang, S., Wu, Z., Su, C., Huang, Q., Tian, Q.: Label decoupling framework for salient object detection. In: CVPR. pp. 13025–13034 (2020) 4
49. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: ECCV. pp. 585–601 (2018) 3, 9, 10, 11
50. Yang, L., Wang, Y., Xiong, X., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: CVPR. pp. 6499–6507 (2018) 3, 9, 10
51. Zeng, X., Liao, R., Gu, L., Xiong, Y., Fidler, S., Urtasun, R.: Dmm-net: Differentiable mask-matching network for video object segmentation. In: ICCV (October 2019) 9, 10
52. Zhang, L., Lin, Z., Zhang, J., Lu, H., He, Y.: Fast video object segmentation via dynamic targeting network. In: ICCV (October 2019) 3, 9
53. Zhao, J.X., Liu, J.J., Fan, D.P., Cao, Y., Yang, J., Cheng, M.M.: Egnet: Edge guidance network for salient object detection. In: ICCV. pp. 8779–8788 (2019) 4
54. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: ICCV (October 2019) 3