READMem: Robust Embedding Association for a Diverse Memory in Unconstrained Video Object Segmentation

Stéphane Vujasinović[1]
stephane.vujasinovic@iosb.fraunhofer.de

Sebastian Bullinger[1]
sebastian.bullinger@iosb.fraunhofer.de

Stefan Becker[1]
stefan.becker@iosb.fraunhofer.de

Norbert Scherer-Negenborn[1]
norbert.scherer-negenborn@iosb.fraunhofer.de

Michael Arens[1]
michael.arens@iosb.fraunhofer.de

Rainer Stiefelhagen[2]
rainer.stiefelhagen@kit.edu

[1] Fraunhofer IOSB[*]
Ettlingen, Germany

[2] Karlsruhe Institut of Technology
Karlsruhe, Germany

Abstract

We present READMem (Robust Embedding Association for a Diverse Memory), a modular framework for semi-automatic video object segmentation (sVOS) methods designed to handle unconstrained videos. Contemporary sVOS works typically aggregate video frames in an ever-expanding memory, demanding high hardware resources for long-term applications. To mitigate memory requirements and prevent near object duplicates (caused by information of adjacent frames), previous methods introduce a hyper-parameter that controls the frequency of frames eligible to be stored. This parameter has to be adjusted according to concrete video properties (such as rapidity of appearance changes and video length) and does not generalize well. Instead, we integrate the embedding of a new frame into the memory only if it increases the diversity of the memory content. Furthermore, we propose a robust association of the embeddings stored in the memory with the query embeddings during the update process. Our approach avoids the accumulation of redundant data, allowing us in return, to restrict the memory size and prevent extreme memory demands in long videos. We extend popular sVOS baselines with READMem, which previously showed limited performance on long videos. Our approach achieves competitive results on the Long-time Video dataset (LV1) while not hindering performance on short sequences. Our code is publicly available at https://github.com/Vujas-Eteph/READMem.
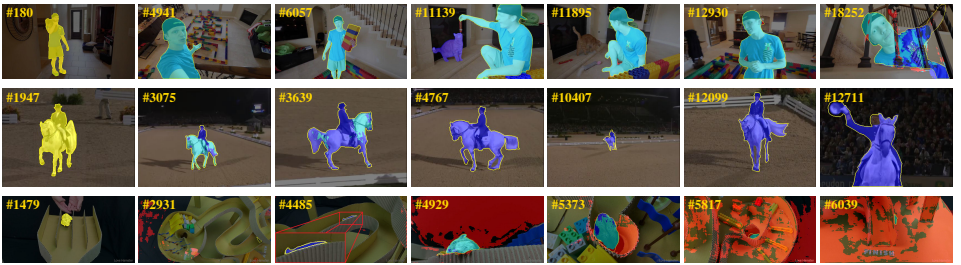
Figure 1: Results of MiVOS [5] (baseline) in red and READMem-MiVOS (READMem added to MiVOS [5]) in blue on the LV1 [17] dataset. We depict the intersection between both predictions in turquoise. We indicate the ground-truth with yellow contours. The first column displays the input frames (*i.e.*, ground-truth mask in yellow) for each sequence of the LV1 [17] dataset used to initialize the methods. The experimental configuration of these results is consistent with the quantitative experiments in Section 4.

1 Introduction

Video object segmentation (VOS), *i.e.*, the pixel-level discrimination of a foreground object from the background in videos, is a fundamental and challenging problem in computer vision. This paper focuses on the semi-automatic video object segmentation (sVOS) setting [44], where the goal consists of segmenting an object (or a collection of objects) in a video by only providing a segmentation mask as input.

Motivation: Current sVOS methods [4, 5, 6, 15, 18, 19, 20] predominantly utilize the space-time memory network design [24], which stores deep representations of previous frames and their estimated masks in an external memory for the segmentation of a query image. Although some attention has recently been given to the memory update strategy [4, 18, 20], most methods rely on a simple frame aggregation approach, which inserts every t-th frame into an ever-expanding memory. This approach works well for short sequences but fails on longer videos due to the saturation of the GPU memory. Furthermore, this strategy neither considers redundancies nor novel information while adding new embeddings to the memory. A naive approach to better deal with long videos is to increase the sampling interval (s_r) of state-of-the-art methods [5, 6, 24], as shown in Figure 2, which displays that increasing the sampling interval leads to better performances. Due to a higher sampling interval, the memory stores embeddings of frames that are further apart in time from each other. These embeddings are more likely to reflect appearance variations resulting in a more diverse set of embeddings in the memory, contributing to better predictions. However, a high sampling interval may lead to the omission of potential useful frames [4, 17], which is particularly detrimental for short sequences. Moreover, as noted by [4], a higher sampling interval leads to unstable performances.

To address the aforementioned points, we propose a simple yet effective extension in the form of a module, named *READMem*, for updating the memory. Drawing inspiration from [28] in the visual object tracking field, our approach focuses on increasing the diversity of the embeddings stored in the memory, rather than simply aggregating every t-th frame. This selective approach allows us to save only key embeddings in the memory, thereby alleviating the memory saturation constraint faced by previous methods on long videos and even on unconstrained video lengths.
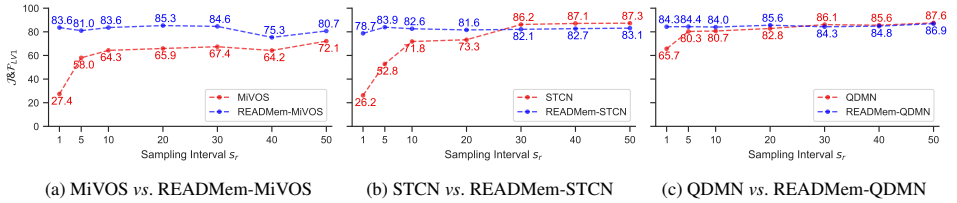
(a) MiVOS *vs.* READMem-MiVOS (b) STCN *vs.* READMem-STCN (c) QDMN *vs.* READMem-QDMN

Figure 2: Performance comparison of sVOS baselines (MiVOS [5], STCN [6], QDMN [20]) with and without the READMem extension on the LV1 [17] dataset when varying the sampling interval s_r. The configuration employed for these experiments is consistent to the quantitative analysis in Section 4.

Our contributions are as follows: (1) We propose READMem, a novel module suitable to extend existing sVOS methods to handle unconstrained videos (arbitrary rapidity of appearance changes and arbitrary video length) without requiring re-training or fine-tuning the sVOS baseline. Our approach improves the diversity of the embeddings stored in the memory and performs a robust association with query embeddings during the update process. (2) When using a small sampling interval (which is desired since this improves the stability of the methods [4] and avoids missing key information [4, 17]) the proposed READMem module consistently improves the results of current sVOS baselines on long videos (LV1 [17]), while not hindering the performance on short videos (*i.e.*, 2017 DAVIS validation set [27] (D17)). Overall, we attain competitive results against the state-of-the-art. (3) We provide insight into our approach through an extensive ablation study. (4) Finally, we make our code publicly available to promote reproducibility and facilitate the implementation of our method.

2 Relate Work

Short-term sVOS Methods: *Online fine-tuning approaches* [3, 21, 22, 32, 34, 35] adapt the network parameters on-the-fly based on an initial mask indicating the object of interest, but suffer from slow inference and poor generalization [44].

In contrast, *propagation-based methods* [10, 12, 13, 26, 43] utilize strong priors offered by previous (adjacent) frames to propagate the mask, allowing them to better deal with fast appearance changes. However, they are prone to error accumulation and lack long-term context for identifying objects after occlusion [44].

Matching-based methods [44] use features of the initial frame along with previously processed frames (intermediate frames) as references for feature matching during inference. In [8, 33, 38, 39, 40, 41], the embeddings of the initial frame and the adjacent frame are matched with the embeddings of the query frame through global and local matching, while Zhou *et al.* [42] also combine online-fine tuning approaches. However, leading sVOS methods rely mostly on the Space-Time Memory (STM) design introduced by Oh *et al.* [24]. Using a non-local approach, the method matches the embeddings from the query frame with the embeddings of reference frames. Based on STM [24], Xie *et al.* [36] and Seong *et al.* [29] perform a local-to-local matching to alleviate distractors-based mismatches. A memory matching at multiple scales is also conducted by Seong *et al.* [30]. To enhance pixel discrimination Yong *et al.* [19] explore a unique approach that uses both, the frequency and spectral domains. To reduce noisy predictions due to the ever-expanding memory, Cheng *et al.* [5]
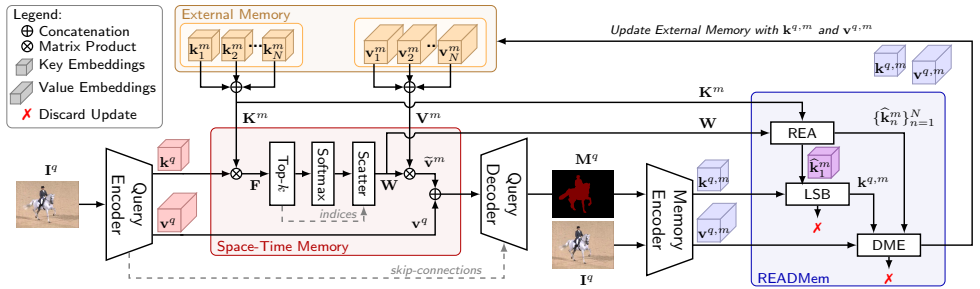
Figure 3: Overview of READMem-MiVOS (MiVOS [5] using READMem) after initialization. We omit the temporary memory for simplicity.

add a top-k filter during non-local matching and propose an efficient extension [6], which decouples the feature extraction of the frame and corresponding mask while improving the matching process. Park *et al.* [25] deviate from the frame-to-frame propagation scheme and adopt instead a clip-to-clip approach to speed-up the propagation process. However, as space-time memory networks-based methods display incredible performance on short-term benchmarks [27, 37], they are hampered in real-world applications by their ever-expanding memory.

Long-term sVOS Methods: To address memory limitations, Li *et al.* [16] propose a compact global module to summarize object segmentation information within a fixed memory size. Similarly, Liang *et al.* [17] apply a weighted average on the extracted features to merge new features with existing ones into the memory and discard obsolete features through a *least frequently used* (LFU) approach. Liu *et al.* [20] explore the use of a quality-aware-module (QAM) [11] to assess the quality of the predicted mask on-the-fly before integrating it in the memory. Li *et al.* [15] use a spatio-temporal aggregation module to update a fixed-sized memory bank. Cheng and Schwing [4] follow the Atkinson-Shiffrin [1] model for their sVOS pipeline (XMem), which comprises: a *sensory memory* [2] that learns an appearance model for every incoming frame, a *working memory* based on STCN [6], and a *long-term memory* that consolidates the working memory embeddings in a compact representation.

3 Method

This section presents our READMem module – an extension for space-time memory-based sVOS pipelines [5, 6, 24]. Figure 3 illustrates our module embedded in MiVOS [5].

3.1 Video Object Segmentation Model Structure

Since READMem is built upon the popular space-time memory network paradigm [24], we provide a brief description of the corresponding components.

(1) A query encoder (an extension of ResNet50 [9]) that encodes a query frame $\mathbf{I}^q$ seen at time step t into a pair of a query key feature $\mathbf{k}^q \in \mathbb{R}^{C_k \times HW}$ and a query value feature $\mathbf{v}^q \in \mathbb{R}^{C_v \times HW}$. With C_k and C_v we denote the number of feature channels, while H and W indicate the spatial image dimensions using a stride of 16.

(2) A memory encoder that extracts latent feature representations from a frame and its corresponding segmentation mask into a pair of a memory key $\mathbf{k}^m \in \mathbb{R}^{C_k \times HW}$ and a memory

value $\mathbf{v}^m \in \mathbb{R}^{C_v \times HW}$. The key features encode visual semantic information that is robust to appearance changes (needed for soft-attention, refer to (4)), while value features encode boundary and texture cues [24] useful at the decoder stage.

(3) An external memory that maintains a set of memory keys $\mathcal{M}^k = \{\mathbf{k}_n^m\}_{n=1}^N$ and a set of memory values $\mathcal{M}^v = \{\mathbf{v}_n^m\}_{n=1}^N$. The variable N denotes the total number of memory slots, while n represents the index of a slot. In addition, some methods [5, 20, 24] use a temporary memory that only contains the latent representation (*i.e.*, memory key and value features) of the previous adjacent frame.

(4) A *Space-Time Memory* (STM) component that serves as an attention mechanism to determine relevant memory information. It matches the channels of the query key $\mathbf{k}^q$ with all channels of every memory key $\mathbf{k}_n^m \in \mathcal{M}^k$ on a temporal and spatial aspect by inferring an affinity matrix $\mathbf{F} \in \mathbb{R}^{NHW \times HW}$. Concretely, the channel level similarities of the query $\mathbf{k}^q$ and every memory key in $\mathbf{K}^m$ are computed through

$$\mathbf{F} = (\mathbf{K}^m)^\mathsf{T}\mathbf{k}^q, \tag{1}$$

where $\mathbf{K}^m \in \mathbb{R}^{C_k \times NHW}$ denotes the matrix obtained from concatenating every $\mathbf{k}_n^m \in \mathcal{M}^k$ along the column dimension, such that $\mathbf{K}^m = \left[\mathbf{k}_{n=1}^m, \mathbf{k}_{n=2}^m, \ldots, \mathbf{k}_{n=N}^m\right]$. To reduce noise (*i.e.*, low affinities) in the affinity matrix $\mathbf{F}$, a top-k filter is applied along the columns (memory dimension) following [5]. This produces a sparse matrix $\mathbf{S} \in \mathbb{R}^{NHW \times HW}$ by

$$\mathbf{S}_{i,j} = \begin{cases} \mathbf{F}_{i,j} & , \text{if } \mathbf{F}_{i,j} \in \operatorname*{argmax}_{\mathcal{F}_j \subset \{f | f \in \mathbf{F}_{*,j}\}, |\mathcal{F}_j| = k} \left(\sum_{f \in \mathcal{F}_j} f \right), \\ 0 & , \text{otherwise} \end{cases} \tag{2}$$

where i and j denote the row and column indices respectively, and $*$ refers to either the complete row or column of a matrix. Based on the sparse affinity matrix $\mathbf{S}$, the method computes soft-weights $\mathbf{W} \in \mathbb{R}^{NHW \times HW}$ using

$$\mathbf{W}_{i,j} = \frac{\exp(\mathbf{S}_{i,j})}{\sum_{l=1}^{NHW} \exp(\mathbf{S}_{l,j})}. \tag{3}$$

All the channels of every memory value feature $\mathbf{v}_n^m \in \mathcal{M}^v$ are weighted with $\mathbf{W}$ to compute a pseudo memory feature representation $\widetilde{\mathbf{v}}^m \in \mathbb{R}^{C_v \times HW}$ through $\widetilde{\mathbf{v}}^m = \mathbf{V}^m \mathbf{W}$, where $\mathbf{V}^m \in \mathbb{R}^{C_k \times NHW}$ is the concatenated form of the memory values $\mathbf{v}_n^m$ along the columns.

(5) Lastly, a decoder inspired by [24] that predicts a segmentation mask $\mathbf{M}^q$ for the query frame $\mathbf{I}^q$ using refinement modules [23], skip-connections and bilinear upsampling.

We use the original weights for the sVOS baselines in our experiments (*i.e.*, MiVOS [5], STCN [6] and QDMN [20]). The training of the sVOS baselines uses a bootstrapped crossentropy loss [5] and consists of two stages: a *pre-training* using static image datasets, and a *main-training* using DAVIS [27] and Youtube-VOS [37]. An intermediate stage can also be performed using the BL30K [5] dataset.

3.2 READMem

Diversification of Memory Embeddings (DME): As mentioned, contemporary methods insert the embeddings of every t-th frame into the memory, along with the corresponding mask. This leads to an inevitable memory overload when accumulating embeddings of

unconstrained sequences. In contrast, READMem proposes an alternative frame insertion scheme, maximizing the diversity of latent representations stored in the memory by only adding frames that enhance information. This allows us to limit the number of memory slots without degrading the performance and avoid GPU memory saturation when processing long videos.

In pursuit of enhancing the diversity of the memory key embeddings $\mathcal{M}^k = \{\mathbf{k}_n^m\}_{n=1}^N$, we require a mean to quantify this diversity. If we conceptually consider the embeddings to form a parallelotope, where the key embeddings $\{\mathbf{k}_n^m\}_{n=1}^N$ act as edges, we can quantify the diversity by calculating the volume of this geometric object. A conventional approach to estimate the volume involves concatenating the keys $\mathbf{k}^m \in \mathbb{R}^{C_k HW}$ (we take a flattened representation of the keys $\mathbf{k}_n^m \in \mathbb{R}^{C_k \times HW}$) into a matrix $\mathbf{X} \in \mathbb{R}^{C_k HW \times N}$ and inferring the determinant of $\mathbf{X}$. However, since $N \ll C_k \times HW$, $\mathbf{X}$ is a non-square matrix, which prevents the computation of the determinant. To circumvent this issue, we leverage the Gram matrix $\mathbf{G} \in \mathbb{R}^{N \times N}$, computed through $\mathbf{G} = \mathbf{X}^{\mathsf{T}} \mathbf{X}$. As demonstrated by [51] (pages 195-196) the determinant of the Gram matrix (*i.e.*, Gramian [51]) allows us to compute the square volume (*vol*) of the parallelotope, such that $(volP(\{\mathbf{k}_n^m\}_{n=1}^N))^2 = \det(\mathbf{G})$. Therefore, we can quantify the diversity of our memory with $\det(\mathbf{G})$. We use the set of memory keys $\mathcal{M}^k = \{\mathbf{k}_n^m\}_{n=1}^N$ to compute the diversity of the memory, as they encode visual semantics that are robust to appearance variation [24]. Hence, by computing the similarity $s_{a,b}$ between the a-th and b-th flatten memory key in $\mathcal{M}^k$ through an inner product $g : \mathbb{R}^{C_k HW} \times \mathbb{R}^{C_k HW} \to \mathbb{R}$ with $g(\mathbf{k}_a^m, \mathbf{k}_b^m) := s_{a,b}$, we construct $\mathbf{G}$ by

$$\mathbf{G}(\mathcal{M}^k) = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N,1} & s_{N,2} & \cdots & s_{N,N} \end{pmatrix}. \tag{4}$$

The size of the memory N is bounded by the dimension of the feature space since otherwise $\det(\mathbf{G}) = 0$ [2]. However, this is not a relevant limitation since in practice, N is several magnitudes smaller than the dimension of the feature space $C_k \times HW$ (*i.e.*, $N \ll C_k \times HW$).

To increase the diversity of the embeddings stored in the memory, we want to maximize the absolute Gramian $|\det(\mathbf{G})|$ of $\mathbf{G}(\mathcal{M}^k)$. As the annotated frame $\mathbf{I}_1$ provides accurate and controlled segmentation information [20, 24], the corresponding memory key $\mathbf{k}_{n=1}^m$ and value $\mathbf{v}_{n=1}^m$ are exempt from the following update strategy:

(1) For each memory slot $n \in \{2, \ldots, N\}$, we substitute the respective memory key $\mathbf{k}_n^m$ with the query-memory key $\mathbf{k}^{q,m}$ (memory key of the current query frame $\mathbf{I}^q$ and mask $\mathbf{M}^q$) and construct a temporary Gram matrix $\mathbf{G}_n^t$. Using the temporary matrices $\mathbf{G}_n^t$, we build a set $\mathcal{G}^t$ containing the absolute values of the determinants, such that $\mathcal{G}^t = \{|\det(\mathbf{G}_n^t)|\}_{n=2}^N$.

(2) The memory is updated based on whether the highest value in $\mathcal{G}^t$ surpasses the absolute value of the current Gramian $|\det(\mathbf{G})|$. If this condition is met, then the corresponding memory slots n (position of the highest value in $\mathcal{G}^t$) is replaced with the query-memory key $\mathbf{k}^{q,m}$ and value $\mathbf{v}^{q,m}$ embeddings.

Robust Embeddings Association (**REA**): Given that we compute an inner-product between two embeddings (enforced by the Gram matrix), we capture the global similarity of two frames in a single score. However, as two frames, $\mathbf{I}_t$ and $\mathbf{I}_{t+\Delta t}$ are further apart in time, it becomes more likely for the object of interest to experience in-between: motion, ego-motion, or size variations. Consequently, a channel-wise embedding may encode different information (specifically foreground *vs.* background) for the same image region in frame $\mathbf{I}_t$ and frame $\mathbf{I}_{t+\Delta t}$ due to the spatial disparity in the object's location. As a result, the
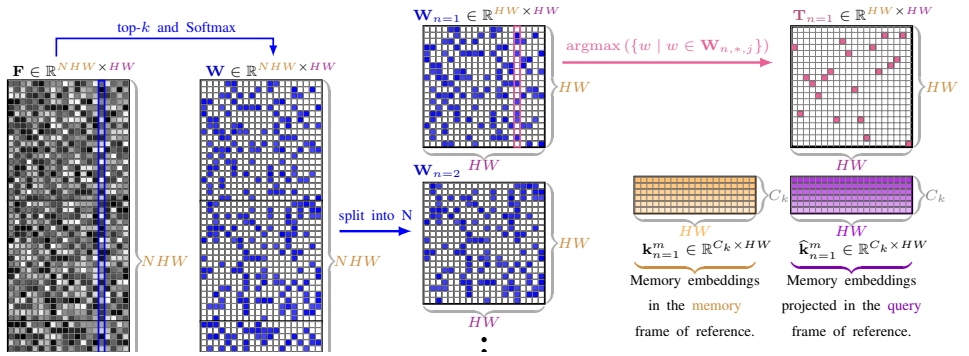
Figure 4: Inference of a transition matrix $\mathbf{T}_n$ and the corresponding pseudo-memory embeddings $\widehat{\mathbf{k}}_n^m$ for robust embeddings association (REA).

resulting similarity between the two frames is inherently low when comparing their embeddings. To address this issue and dampen positional variation without relying on a motion model or fixed-sized search region (which would introduce additional hyper-parameters), our proposal utilizes transition matrices $\{\mathbf{T}_n\}_{n=1}^N$. Wherein, the core idea is to leverage the cross-attention already at hand in the memory-based networks (*i.e.*, $\mathbf{W}$ and $\mathbf{F}$) to identify the best channel-wise mapping between two embeddings. Specifically, through the transition matrix $\mathbf{T}_n \in \mathbb{R}^{HW \times HW}$, we project the n-th memory embedding $\mathbf{k}_n^m \in \mathbb{R}^{C_v \times HW}$ from its original *frame of reference* (*FoR*) to the query's *FoR* (see Figure 4) by

$$\widehat{\mathbf{k}}_n^m = \mathbf{k}_n^m \mathbf{T}_n, \tag{5}$$

where $\widehat{\mathbf{k}}_n^m \in \mathbb{R}^{C_v \times HW}$ denotes the pseudo-embedding of the embedding $\mathbf{k}_n^m$, but expressed in the query's *frame of reference*. This approach effectively compensates for the target's spatial disparity between two distant frames (refer to Table 2 and 3).

Thus, we compute the similarity w.r.t. $\mathbf{k}^{q,m}$ not with the memory keys $\{\mathbf{k}_n^m\}_{n=1}^N$ but with the pseudo memory keys $\{\widehat{\mathbf{k}}_n^m\}_{n=1}^N$. Figure 4 depicts the operation that maps key embeddings from the memory's *FoR* to the query's *FoR* through the transition matrix $\mathbf{T}_n$. We use $\mathbf{W}$, a filtered version of $\mathbf{F}$ (see (3)), to only consider strong affinities between the point-wise embeddings (channels) of the query and memory key which are more likely to encode a similar information. This ensures that the corresponding mapping matrix $\mathbf{T}_n$ is constructed based on relevant channels (refer to Table 3). Although $\mathbf{W}$ entails strong similarities, using $\mathbf{W}_n$ as transition matrix $\mathbf{T}_n$ potentially leads to the aggregation of multiple memory channels onto a single pseudo memory channel – which is certain to degrade the similarity. Hence, to avoid the summation, we need to map at most one element from the memory *FoR* to the query *FoR*, *i.e.*, filtering $\mathbf{W}_n$. Generating a bijective mapping between the memory key $\mathbf{k}_n^m$ and the corresponding pseudo query key $\widehat{\mathbf{k}}_n^m$ would constrain the foreground-background ratio of $\widehat{\mathbf{k}}_n^m$ based on $\mathbf{k}_n^m$. However, it is unlikely that the area taken by the object in the image is unchanged from one frame to the other. To avoid this foreground-background restriction, we should allow point-wise embeddings from the memory key $\mathbf{k}_n^m$ to be re-used for the creation of the pseudo memory key $\widehat{\mathbf{k}}_n^m$, as long as they are sufficiently similar. Thus, an appropriate function that validates the aforementioned criteria is argmax applied along the columns of $\mathbf{W}$ to maximize the re-usability of point-wise embeddings when producing the pseudo memory key $\widehat{\mathbf{k}}_n^m$ on the query *FoR*. We obtain $\mathbf{T}_n \in \mathbb{R}^{HW \times HW}$ by dividing $\mathbf{W} \in \mathbb{R}^{NHW \times HW}$ into N-

square matrices, such that $\mathbf{W}_n \in \mathbb{R}^{HW \times HW}$, and by applying argmax along the columns by

$$\mathbf{T}_{n,i,j} = \begin{cases} 1 & \text{, if } i \in \text{argmax}\left(\{w \mid w \in \mathbf{W}_{n,*,j}\}\right) \\ 0 & \text{, otherwise} \end{cases} . \tag{6}$$

Lower Similarity Bound of Memory Embeddings (**LSB**): To ensure the presence of the foreground object in a candidate frame, we set a lower bound on similarity $l_{bs} = 0.5$. We validate object presence in $\mathbf{I}_q$ by computing a similarity score such that $g(\widehat{\mathbf{k}}_1^m, \mathbf{k}^{q,m}) > l_{bs}$.

Initialization: We integrate every t-th frame (that satisfies the lower similarity bound) into the memory – until the memory is completely filled. Afterward, the method only integrates embeddings of frames that enhance the diversity of the memory.

4 Experiments

In our experiments, we use the Long-time Video [[17]] (LV1) dataset (long videos) as well as the validation set of the DAVIS 2017 [[27]] (D17) dataset (short videos). We use the $\mathcal{J}$ (intersection over union) and $\mathcal{F}$ (contour accuracy) metrics (higher is better) introduced by [[27]] to quantify the performance of each method. Both metrics are averaged into the $\mathcal{J}\&\mathcal{F}$ metric for ease of ranking. We perform all experiments on an Nvidia GeForce GTX 1080 Ti.

Quantitative Results: Table 1 presents the quantitative results of recent sVOS methods (MiVOS [[5]], STCN [[6]] and QDMN [[20]]) and their READMem based extensions along with the state-of-the-art sVOS method XMem [[4]] on the LV1 [[17]] and the D17 [[27]] datasets – we display the qualitative results of MiVOS [[5]] and READMem-MiVOS on LV1 [[17]] in Figure 1.

To ensure a fair comparison, we keep the default settings for XMem [[4]] and use the same sampling interval (*i.e.*, $s_r = 10$) for all other methods (in contrast to previous works [[4], [17]] that use dataset-specific sampling intervals). In our experiments, we follow the evaluation of [[20]] and limit the number of memory slots of the baselines and the corresponding READMem counterparts to 20. Note that this value differs from the memory slot limit (*i.e.*, 50) used by [[4], [17]]. In contrast to [[4], [17]], we do not adapt the sampling interval to the sequence length when dealing with long videos as we argue that the sampling interval should not be correlated to the video's length and is not a standard practice when dealing with short videos (D17 [[27]] and Y-VOS [[37]]). Instead, we opt for a first-in-first-out (*i.e.*, FIFO) replacement strategy when updating the memory of the mentioned baselines. Our results in Table 1 demonstrate that READMem improves long-term performance while preserving good short-term capabilities. Moreover, by setting $s_r = 1$ we minimize the likelihood of omitting key frames and enhance the stability of the method [[4]]. As a result, we attain competitive results against the state-of-the-art method (*i.e.*, XMem [[4]]). We do not employ READMem on XMem [[4]], as a long-term memory is already present, consolidating the embeddings of the working memory [[6]] and updated by a *least-frequently-used* (*LFU*) approach [[4], [17]].

Ablation Studies: For clarity and to isolate the benefits of each design decision, we present our ablation results solely for READMem-MiVOS in Table 2. We observe that integrating embeddings of frames that diversify the memory (DME) and enforcing a lower bound on similarity (LSB) results in a significant performance improvement for long videos. Moreover, using the robust association of memory embeddings (REA) also leads to a substantial performance increase. Although including the adjacent frame leads to a slight improvement for long videos, it is particularly important when dealing with short sequences as the previous adjacent frame provides recent information (not guaranteed by our module).

Method	LV1 [17] (Long Sequences)			D17 [27] (Short Sequences)		
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Sampling Interval $s_r = 10$						
XMem† [4] (ECCV 22)	**89.8**	**88.0**	**91.6**	85.5	82.3	88.6
XMem [4] (ECCV 22)	83.6	81.8	85.4	**86.7**	83.0	**90.1**
MiVOS [5] (CVPR 21)	64.3	63.6	65.0	84.3	81.4	87.1
READMem-MiVOS (ours)	$83.6^{\uparrow19.3}$	$82.1^{\uparrow18.5}$	$85.1^{\uparrow20.1}$	84.3	81.4	87.1
READMem-MiVOS* (ours)	$86.0^{\uparrow21.7}$	$84.6^{\uparrow21.0}$	$\underline{87.4}^{\uparrow22.4}$	$84.6^{\uparrow0.3}$	$81.8^{\uparrow0.4}$	$87.3^{\uparrow0.2}$
STCN† [6] (NIPS 21)	71.8	69.4	74.3	83.7	80.7	86.6
READMem-STCN† (ours)	$82.6^{\uparrow10.8}$	$81.5^{\uparrow12.1}$	$83.7^{\uparrow9.4}$	83.7	80.7	86.6
READMem-STCN†* (ours)	$81.8^{\uparrow10.0}$	$80.8^{\uparrow11.4}$	$82.8^{\uparrow8.5}$	83.7	80.7	86.6
QDMN* [20] (ECCV 22)	80.7	77.8	83.6	86.0	$\underline{83.2}$	88.8
READMem-QDMN* (ours)	$84.0^{\uparrow3.3}$	$81.3^{\uparrow3.5}$	$86.7^{\uparrow3.1}$	$86.1^{\uparrow0.1}$	$\mathbf{83.3}^{\uparrow0.1}$	$\underline{88.9}^{\uparrow0.1}$
Sampling Interval $s_r = 1$						
XMem† [4] (ECCV 22)	61.5	60.4	68.4	84.1	81.1	**90.8**
XMem [4] (ECCV 22)	79.3	77.3	84.2	83.9	80.6	87.1
MiVOS [5] (CVPR 21)	27.4	27.4	27.4	84.3	81.3	87.3
READMem-MiVOS (ours)	$\underline{83.6}^{\uparrow56.2}$	$82.3^{\uparrow54.9}$	$\underline{85.0}^{\uparrow57.6}$	84.3	$81.4^{\uparrow0.1}$	$87.1^{\downarrow0.2}$
STCN† (NIPS 21)	26.2	22.9	29.4	83.2	79.9	86.5
READMem-STCN† (ours)	$80.8^{\uparrow54.6}$	$78.4^{\uparrow55.5}$	$83.2^{\uparrow53.8}$	$83.8^{\uparrow0.6}$	$80.4^{\uparrow0.5}$	$87.2^{\uparrow0.7}$
QDMN (ECCV 22)	65.7	63.5	67.9	$\underline{85.1}$	82.2	88.0
READMem-QDMN (ours)	$84.3^{\uparrow18.6}$	$81.9^{\uparrow18.4}$	$86.7^{\uparrow18.8}$	$85.3^{\uparrow0.2}$	$82.4^{\uparrow0.2}$	$88.1^{\uparrow0.1}$

Table 1: Quantitative evaluation of sVOS methods [4, 5, 6, 20] with and without READMem on the LV1 [17] and D17 [27] datasets. The symbol † denotes no pre-training on BL30K [5], while * indicates the use of a flexible sampling interval as in QDMN [20].

Configuration	$\mathcal{J}\&\mathcal{F}_{\text{LV1}}$	$\mathcal{J}\&\mathcal{F}_{\text{D17}}$
MiVOS [5] + adj. frame (baseline)	64.3	84.3
MiVOS [5] + DME (ours)	$69.5^{\uparrow5.2}$	$81.3^{\downarrow3.0}$
MiVOS [5] + DME + LSB (ours)	$75.0^{\uparrow10.7}$	$81.3^{\downarrow3.0}$
MiVOS [5] + DME + LSB + adj. frame (ours)	$77.4^{\uparrow13.1}$	$84.3^{\uparrow0.0}$
MiVOS [5] + DME + LSB + adj. frame + REA (ours)	$\mathbf{86.0}^{\uparrow21.7}$	$84.6^{\uparrow0.3}$

Table 2: Demonstrating the benefits of our memory extension, *i.e.*, diversity of the embeddings (DME), robust embedding association (REA) and lower similiraty bound (LSB).

Computation of the transition matrix $\mathbf{T}_n$	$\mathcal{J}\&\mathcal{F}_{\text{LV1}}$ using $\mathbf{W}_n$	$\mathcal{J}\&\mathcal{F}_{\text{LV1}}$ using $\mathbf{F}_n$
Hungarian Method [14]	76.8	75.1
argmax along the columns	**86.0**	73.4
argmax along the rows	79.8	**77.1**

Table 3: Performance comparison on the LV1 [17] dataset when computing the transition matrix $\mathbf{T}_n$ through different methods. As a reminder, the baseline (*i.e.*, MiVOS [5]) achieves a $\mathcal{J}\&\mathcal{F}$ score of 64.3 on LV1 [17] using the same configuration.

	blueboy		*dressage*		*rat*	
s_r	$\mathcal{J}\&\mathcal{F}$	$|\mathbf{G}|$	$\mathcal{J}\&\mathcal{F}$	$|\mathbf{G}|$	$\mathcal{J}\&\mathcal{F}$	$|\mathbf{G}|$
1	**89.5**	14.6×10^{-7}	83.7	3.36×10^{-8}	77.7	4.22×10^{-10}
5	87.0	6.35×10^{-7}	83.7	9.38×10^{-8}	74.9	1.39×10^{-10}
10	88.1	4.31×10^{-7}	**84.0**	$\mathbf{18.0 \times 10^{-8}}$	**86.0**	$\mathbf{82.3 \times 10^{-10}}$

Table 4: Performance and Gramian of READMEm-MiVOS for different sampling intervals s_r on LV1 [17]. Note that high $\mathcal{J}\&\mathcal{F}$ scores correlate with high Gramian values.
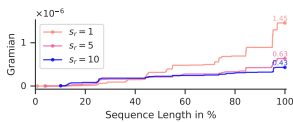
Figure 5: Evolution of $|\mathbf{G}|$ for different sampling interval on *blueboy* [17].

s_r	0.0% − 26.20%		26.20% − 57.64%		57.64% − 83.63%	
	$\mathcal{J}\&\mathcal{F}$	$|\mathbf{G}|$	$\mathcal{J}\&\mathcal{F}$	$|\mathbf{G}|$	$\mathcal{J}\&\mathcal{F}$	$|\mathbf{G}|$
1	**94.3**	0.61×10^{-7}	**72.5**	$\mathbf{4.80 \times 10^{-7}}$	95.6	$\mathbf{6.93 \times 10^{-7}}$
5	93.3	1.46×10^{-7}	59.3	2.34×10^{-7}	92.2	3.45×10^{-7}
10	90.4	$\mathbf{1.69 \times 10^{-7}}$	66.1	2.44×10^{-7}	95.4	3.09×10^{-7}

Table 5: Relative performance and Gramian (*i.e.*, $|\mathbf{G}|$) evolution for three sections on the *blueboy* sequence [17]. The final Gramian and $\mathcal{J}\&\mathcal{F}$ score is reported in Table 4.

In Table 3, we compare different approaches for inferring the transition matrix $\mathbf{T}_n$. Our results indicate, as expected, that using the weight matrix $\mathbf{W}$ to generate the transition matrices $\mathbf{T}_n$ leads to better performance compared to the affinity matrix $\mathbf{F}$. Furthermore, we note that using the argmax function along the columns axis (memory) leads to the best performance in comparison to applying argmax along the rows axis (query) or when using a bijective mapping (Hungarian Method [14]).

Table 4 tabulates the $\mathcal{J}\&\mathcal{F}$ score and Gramian of READMem-MiVOS on the three sequences of LV1 [17] for three different sampling interval s_r. In Figure 5, we display the evolution of the Gramian over the observed *blueboy* sequence along with the specific attained Gramian and the respective $\mathcal{J}\&\mathcal{F}$ performance of three intermediate sections in Table 5. We note that the Gramian continuously increases over the observed sequences and provides more assistance in the latter stages as indicated in Table 5, where a high Gramian generally correlates to higher performance.

5 Conclusion

We propose READMem, a modular framework for STM-based sVOS methods, to improve the embeddings-stored in the memory on unconstrained videos and address the expanding memory demand. Our evaluation displays that sVOS methods [5, 6, 20] using our extension consistently improve their performance compared to their baseline on the long video sequences (*i.e.*, LV1 [17]) without compromising their efficiency on shorter sequences (*i.e.*, D17 [22]). Moreover, we achieve competitive results with state-of-the-art and provide a comprehensive overview of each design decision, supported by an extensive ablation study.

References

[1] Richard C Atkinson and Richard M Shiffrin. Human memory: A proposed system and its control processes. In *Psychology of learning and motivation*. Elsevier, 1968.

[2] Nils Barth. The gramian and k-volume in n-space: some classical results in linear algebra. *Journal of Young Investigators*, 1999.

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision (ECCV)*, 2022.

[5] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[6] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[8] Suhwan Cho, Heansung Lee, Minhyeok Lee, Chaewon Park, Sungjun Jang, Minjung Kim, and Sangyoun Lee. Tackling background distraction in video object segmentation. In *European Conference on Computer Vision (ECCV)*, 2022.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Ping Hu, Gang Wang, Xiangfei Kong, Jason Kuen, and Yap-Peng Tan. Motion-guided cascaded refinement network for video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask Scoring R-CNN. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955.

[15] Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu. Recurrent dynamic embedding for video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[16] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *European Conference on Computer Vision (ECCV)*, 2020.

[17] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[18] Zhihui Lin, Tianyu Yang, Maomao Li, Ziyu Wang, Chun Yuan, Wenhao Jiang, and Wei Liu. Swem: Towards real-time video object segmentation with sequential weighted expectation-maximization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[19] Yong Liu, Ran Yu, Jiahao Wang, Xinyuan Zhao, Yitong Wang, Yansong Tang, and Yujiu Yang. Global spectral filter memory network for video object segmentation. In *European Conference on Computer Vision (ECCV)*, 2022.

[20] Yong Liu, Ran Yu, Fei Yin, Xinyuan Zhao, Wei Zhao, Weihao Xia, and Yujiu Yang. Learning quality-aware dynamic memory for video object segmentation. In *European Conference on Computer Vision (ECCV)*, 2022.

[21] 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. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[22] Tim Meinhardt and Laura Leal-Taixé. Make one-shot video object segmentation efficient again. *Neural Information Processing Systems (NeurIPS)*, 2020.

[23] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *International Conference on Computer Vision (ICCV)*, 2019.

[25] Kwanyong Park, Sanghyun Woo, Seoung Wug Oh, In So Kweon, and Joon-Young Lee. Per-clip video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[26] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

[28] Axel Sauer, Elie Aljalbout, and Sami Haddadin. Tracking holistic object representations. In *British Machine Vision Conference (BMVC)*, 2019.

[29] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *European Conference on Computer Vision (ECCV)*, 2020.

[30] Hongje Seong, Seoung Wug Oh, Joon-Young Lee, Seongwon Lee, Suhyeon Lee, and Euntai Kim. Hierarchical memory matching network for video object segmentation. In *International Conference on Computer Vision (ICCV)*, 2021.

[31] Èrnest Borisovich Vinberg. *A course in algebra*. American Mathematical Soc., 2003.

[32] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *British Machine Vision Conference (BMVC)*, 2017.

[33] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[35] Huaxin Xiao, Bingyi Kang, Yu Liu, Maojun Zhang, and Jiashi Feng. Online meta adaptation for fast video object segmentation. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[36] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[37] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[38] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[39] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *European Conference on Computer Vision (ECCV)*, 2020.

[40] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[41] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[42] Mao Yunyao, Wang Ning, Zhou Wengang, and Li Houqiang. Joint inductive and transductive learning for video object segmentation. In *International Conference on Computer Vision (ICCV)*, 2021.

[43] Lu Zhang, Zhe Lin, Jianming Zhang, Huchuan Lu, and You He. Fast video object segmentation via dynamic targeting network. In *International Conference on Computer Vision (ICCV)*, 2019.

[44] Tianfei Zhou, Fatih Porikli, David J Crandall, Luc Van Gool, and Wenguan Wang. A survey on deep learning technique for video segmentation. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.