Supplementary Material

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

Stéphane Vujasinović
stephane.vujasinovic@iosb.fraunhofer.de

Sebastian Bullinger
sebastian.bullinger@iosb.fraunhofer.de

Stefan Becker
stefan.becker@iosb.fraunhofer.de

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

Michael Arens
michael.arens@iosb.fraunhofer.de

Rainer Stiefelhagen
rainer.stiefelhagen@kit.edu

1 Fraunhofer IOSB*
Ettlingen, Germany

2 Karlsruhe Institute of Technology
Karlsruhe, Germany

In this supplementary document, we provide additional experiments, visualizations and insights.

A Additional Qualitative Results on the Long-time Video (LV1 [8]) Dataset

We display qualitative results for the READMem variations of MiVOS [9], STCN [8] and QDMN [9] along with their baseline on the LV1 [8] dataset. We use the same settings as described in Section 4 (refer to quantitative results). We also provide the results for XMem [8], which represents the state-of-the-art.

Figures S1, S2 and S3 displays the results for the blueboy, dressage and rat sequences in LV1 [8] respectively when using \( s_r = 10 \), while Figures S4, S5 and S6 display the results for \( s_r = 1 \). The estimated segmentation mask of the baselines (MiVOS [9], STCN [8], and QDMN [9]) are visualized in red, while the results of the READMem-based variations (READMem with a baseline) are highlighted in blue. The intersection between the prediction of a baseline and its corresponding READMem variation is depicted in turquoise. The ground-truth contours are highlighted in yellow. We depict XMem [8] results in purple.

© 2023. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

*Fraunhofer IOSB is a member of the Fraunhofer Center for Machine Learning.
A.1 Qualitative Results on LV1 [8] with $s_r = 10$

Figure S1: Results on the blueboy sequence of LV1 [8] with $s_r = 10$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

Figure S2: Results on the dressage sequence of LV1 [8] with $s_r = 10$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

Figure S3: Results on the rat sequence of LV1 [8] with $s_r = 10$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.
A.2 Qualitative Results on LV1 [8] with $s_r = 1$

Figure S4: Results on the blueboy sequence of LV1 [8] with $s_r = 1$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

Figure S5: Results on the dressage sequence of LV1 [8] with $s_r = 1$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

Figure S6: Results on the rat sequence of LV1 [8] with $s_r = 1$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.
### B Additional Quantitative Evaluation

We present in Table S1 useful statistics for popular (sVOS) and Visual Object Tracking (VOT) datasets. As our goal is to allow sVOS methods to perform on long video sequences, Table S1 reveals that the LV1 dataset and the recently introduced VOTS2023 dataset are ideal candidates for assessing the effectiveness of our READMem extension.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of sequences</th>
<th>avg. length</th>
<th>median length</th>
<th>std.</th>
<th>min. length</th>
<th>max. length</th>
</tr>
</thead>
<tbody>
<tr>
<td>D17 (validation set)</td>
<td>30</td>
<td>67</td>
<td>67</td>
<td>21</td>
<td>34</td>
<td>104</td>
</tr>
<tr>
<td>YVOS (validation set)</td>
<td>507</td>
<td>134</td>
<td>144</td>
<td>42</td>
<td>16</td>
<td>180</td>
</tr>
<tr>
<td>LV1</td>
<td>3</td>
<td>2470</td>
<td>2406</td>
<td>1088</td>
<td>1416</td>
<td>3589</td>
</tr>
<tr>
<td>VOT2022</td>
<td>62</td>
<td>321</td>
<td>242</td>
<td>295</td>
<td>41</td>
<td>1500</td>
</tr>
<tr>
<td>VOTS2023</td>
<td>144</td>
<td>2073</td>
<td>1810</td>
<td>1856</td>
<td>63</td>
<td>10699</td>
</tr>
</tbody>
</table>

Table S1: Statistics of popular sVOS and VOT datasets. For more details refer to the original publications.

#### B.1 Performance on the DAVIS (D17) Dataset

We display the performance of MiVOS, STCN and QDMN with and without the READMem extension when varying the sampling interval $s_r$ on the D17 dataset, using the same configuration as in Section 4.

In Figure 2 of Section 1, we observe that increasing the sampling interval generally improves the performance of all methods on long videos, regardless of the baseline employed. However, this trend does not hold when working with short video sequences, as shown in Figure S7. Here, we notice a degradation in performance for all methods when using larger sampling intervals.

Figure S7: Performance comparison of sVOS baselines (MiVOS, STCN, QDMN) with and without the READMem extension on the D17 dataset, while varying the sampling interval $s_r$. Regardless of the final performance, we observe a general tendency where increasing the sampling interval (i.e., $s_r$ higher than 10) on short video sequences leads to a performance drop.
Therefore, it is essential to utilize a sampling interval that does not negatively impact the performance on both long and short video sequences. This is where our READMem extension becomes valuable, as it enables the sVOS pipeline to use a small sampling interval (typically \( s_r \in [1 - 10] \)) that achieves and maintains high performance for both long and short video sequences.

### B.2 Performance as a Function of Memory Size

We explore the impact of the size of the memory on the performance of MiVOS [1], STCN [2] and QDMN [3] with and without our READMem extension on the LV1 [4] dataset. We follow the same experimental setup as in Section 4 (with \( s_r = 10 \)), except for the varying memory size \( N \), which ranges from 5 to 50.

From Figure S8, we observe that the performance of the baselines improves as the memory size increases. Similarly, although to a lesser extent, the READMem variants also demonstrate improved performance with larger memory sizes. However, the READMem variations consistently outperform their respective baselines, especially when using a smaller memory size. This is desired as a smaller memory requires less GPU resources.

Comparing Figure S8 with Figure 2, we notice that increasing the sampling interval (i.e., \( s_r \)) of the baselines leads to a significant boost in performance compared to increasing the memory size (i.e., \( N \)). Hence, storing a diverse set of embeddings in the memory is more beneficial than including additional ones.

### B.3 Performance on the VOTS2023 [5] Dataset

In our quantitative evaluation (refer to Table 1 of Section 4), we demonstrate and analyze the effectiveness of our approach on sVOS datasets, encompassing both short (i.e., D17 [6]) and long (i.e., LV1 [4]) sequences, to allow for a direct comparison with contemporary sVOS approaches (i.e., [4, 6]). In an effort, to enhance the soundness of our READMem extension, we conduct additional experiments on the VOTS2023 dataset [5]. We tabulate in Table S2, the results of sVOS baselines [1, 2, 3] with and without READMem on the VOTS2023 tracking benchmark.

For the evaluation we use the same settings as described in Section 4 (refer to quantitative results) and the official VOT evaluation toolkit (version 0.6.4 released on the 31 May 2023 – https://github.com/votchallenge/toolkit). We observe from Table S2, that the READMem variants consistently outperform their baseline counterpart.

Figure S8: We compare the performance of sVOS baselines (MiVOS [1], STCN [2], QDMN [3]) with and without the READMem extension on the LV1 [4] dataset while varying the size of the memory (i.e., \( N \)). A general tendency is that increasing the memory size, leads to better performance.
D Discussion and Limitations

We are aware of the limitations imposed by the hand-crafted threshold for the lower similarity bound $l_{sb}$, although to avoid any fine-tuning, we set the threshold value to 0.5. A

<table>
<thead>
<tr>
<th>Initialization</th>
<th>READMem-MiVOS $J &amp; F_{LV1}$</th>
<th>$J &amp; F_{D17}$</th>
<th>READMem-STCN $J &amp; F_{LV1}$</th>
<th>$J &amp; F_{D17}$</th>
<th>READMem-QDMN $J &amp; F_{LV1}$</th>
<th>$J &amp; F_{D17}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>83.6</td>
<td>84.6</td>
<td>82.6</td>
<td>84.0</td>
<td>84.0</td>
<td>86.1</td>
</tr>
<tr>
<td>(2)</td>
<td>82.7</td>
<td>73.7</td>
<td>85.3</td>
<td>73.6</td>
<td>72.5</td>
<td>73.3</td>
</tr>
</tbody>
</table>

Table S3: Performance variation when leveraging two different initialization strategies for READMem-MiVOS. Besides the initialization strategy, the remaining parameters are consistent to Section 4 (we set $s_r = 10$).
more thoughtful approach would incorporate a learnable parameter. This approach could potentially lead to improved performance, albeit at the expense of the plug-and-play nature of our extension. Another point for improvement is to reduce the participation of the background when computing the similarity between two embeddings. A possible enhancement is to integrate either the segmentation mask estimated by the sVOS pipeline or use the memory values to estimate a filter that can be applied to the memory keys before computing a similarity score.

E Training

For our experiments, we utilize the original weights provided by the authors of MiVOS [2], STCN [3], and QDMN [9]. Our primary focus is to showcase the benefits of our extension (i.e., READMem) without modifying the baselines. To provide a comprehensive overview of the baselines, we briefly elaborate on the training methodology. The training procedure follows the regiment presented in STM [10] and refined in the subsequent work, MiVOS [2]. The training is divided into two stages employing the bootstrapped cross-entropy loss [2] and utilizing the Adam optimizer (refer to the original papers [2, 3, 9] and their supplementary materials for detailed insights).

The training comprises the following stages: (1) A pre-training stage, in which static image datasets are used as in [2] to simulate videos consisting of three frames. While all three frames originate from the same image, the second and third frames are modified using random affine transformations (2) A main-training stage, which uses the DAVIS [2] and the Youtube-VOS [3] datasets (which provide real videos). Similar to the pre-training stage, three frames from a video are sampled, gradually increasing the temporal gap from 5 to 25 frames during training. Subsequently, the temporal gap is annealed back to 5 frames, following a curriculum training approach [2]. (Optional) Moreover, after the pre-training stage a synthetic dataset BL30K [2] can be leveraged to enhance the ability of the model to better handle complex occlusion patterns.

References


