Alignment Before Aggregation: Trajectory Memory Retrieval Network for Video Object Segmentation

Supplementary Material

Rui Sun[1*] Yuan Wang[1*] Huayu Mai[1] Tianzhu Zhang[1,2,3†] Feng Wu[1,2]
[1] University of Science and Technology of China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[3]Deep Space Exploration Lab
{issunrui,wy2016,mai556}@mail.ustc.edu.cn, {tzzhang, fengwu}@ustc.edu.cn

In the supplementary material, we first introduce more details about the network architecture of the three baselines including memory frame encoder (*i.e.*, $\mathbf{M}^k$, $\mathbf{M}^v$ as output), query frame encoder (*i.e.*, $\mathbf{Q}^k$, $\mathbf{Q}^v$ as output) and the decoder (*i.e.*, **Decoder** in Figure 1). Then, we elaborate the implementation details of the three baselines including the similarity function and the training strategies (as a supplement to Section 4.1 in main paper). Finally, we showcase more qualitative results of our TMRN (as a supplement to Section 4.2).

1. More Details on Network Architecture

In this section, we provide more details about the network architecture of the three baselines (*i.e.*, STM [9], XMem [2] and STCN [3]), including memory frame encoder (*i.e.*, $\mathbf{M}^k$, $\mathbf{M}^v$ as output), query frame encoder (*i.e.*, $\mathbf{Q}^k$, $\mathbf{Q}^v$ as output) and the decoder (*i.e.*, **Decoder** in Figure 1).

For STM, we prepend the TMRN to improve the memory reading module (MRM). The query encoder takes the query frame as the input, and outputs two feature maps including key $\mathbf{Q}^k$ and value $\mathbf{Q}^v$ through two parallel convolutional layers attached to the backbone network ResNet50 [4]. The structure of the memory encoder is the same as the query encoder except that the input is expanded to four channels including the RGB frame and the segmented mask, and the output is the memory key $\mathbf{M}^k$ and value $\mathbf{M}^v$ from `res4` features with stride 16. The **decoder** takes the concatenation of retrieved memory value $\mathbf{v}$ (Section 3.2) and the query value $\mathbf{Q}^v$ and compresses them into 256 channels through a convolutional layer and the residual block, and then a series of refinement modules gradually upsample the compressed features by a factor of two at a time to attain the final prediction.

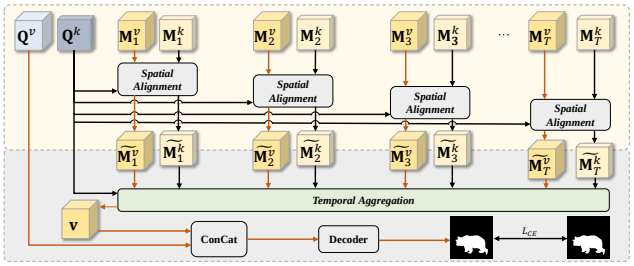

Figure 1. Illustration of the proposed TMRN. TMRN is mainly composed of a spatial alignment module and a temporal aggregation module to equip with the trajectory information, and enables each query pixel to independently retrieve the pixels in each memory frame to seek the location of the counterpart trajectory, and obtain spatially aligned memory pixel features. Then the resultant aligned memory pixels are pooled through the temporal aggregation module to reason about inter-frame connections.

For STCN, we develop the TMRN to improve the memory reading module (MRM). Unlike STM, a key encoder (image as input, and output memory key $\mathbf{M}^k$, query key $\mathbf{Q}^k$) and a value encoder (image and mask as input, and output memory value $\mathbf{M}^v$) are constructed with ResNet50 and ResNet18 respectively. Note that the query value $\mathbf{Q}^v$ is obtained by applying another convolution layer after the key encoder which takes the current frame as input. The **decoder** structure stays close to that of the STM, that is, features are processed and upsampled at a scale of two gradually with higher-resolution features incorporated credited to skip-connections. Then the final layer of the decoder produces a stride 4 mask which is bilinearly upsampled to the original resolution.

For XMem, we integrate TMRN into working memory and disable long-term memory. Disabling long-term memo-

*Equal contribution
†Corresponding author

ry can make the model work better on DAVIS and Youtube-VOS variant benchmarks, as indicated in [2]. Similar to STCN, the key encoder (memory key $\mathbf{M}^k$, query key $\mathbf{Q}^k$) is on top of ResNet50 and value encoder (memory value $\mathbf{M}^v$ as output) is based on ResNet18 respectively. And the query value $\mathbf{Q}^v$ is obtained by applying another convolution layer after the key encoder. The decoder concatenates hidden representation from sensory memory and retrieved features. Then these resultant features are iteratively upsampled by $2\times$ at a time until stride 4 and projected to a single channel logit via a 3×3 convolution for segmentation.

2. More on Implementation Details

In this section, we elaborate the implementation details of the three baselines (*i.e.*, STM [9], XMem [2] and STCN [3]) including the similarity function (Section 3.2) and the training strategies. Note that all the rest of the network architecture (*i.e.*, except TMRN) including memory frame encoder and query frame encoder, and training settings are exactly the same as the baselines. Generally, all baselines undergo two-stage training, including static image pretraining [1, 6, 11, 12, 14] and video data main training [10, 13]. For STM, we take the dot product as the similarity function, and crop 384×384 for training, and minimize the cross-entropy loss using the Adam [5] optimizer with an initial learning rate of 1e-5. And for STCN, we utilize the negative squared Euclidean distance as similarity function. During main training, three frames are randomly sampled from a video, and the sampling interval gradually increases from 5 to 25 as a curriculum leaning schedule and anneals back to 5 towards the end of training. We use a batch size of 16 during pretraining and a batch size of 8 during main training with the bootstrapped cross entropy. For XMem, we devise anisotropic L2 by introducing two scaling terms as similarity function. As implemented in XMem, we sample sequences of length eight, and a maximum of 3 past frames are randomly selected to be the working memory for reducing training time. For optimization, we adopt the AdamW [8] optimizer with the learning rate of 1e-5 based on the bootstrapped cross-entropy loss combined with the dice loss. During inference, we construct memory frames with a sampling interval of 5, and employ the soft aggregation operation when multiple target objects exist in a video.

3. More Qualitative Results

Figure 2 showcases qualitative comparison between STCN w/ TMRN and other competitive methods including STM [9], GSFM [7], and STCN [3] on YouTube-VOS [13]. We can vividly observe that GSFM and STCN fail to predict target objects when multiple similar objects *zebra* have appeared (*i.e.*, *1ab5f4bbc5* in Figure 2), while our TMRN can accurately discriminate the distractors. This is in line with the design idea of agent-level correlation, that is, alleviating false matches caused by direct pixel-level correlation and pursuing that true pixel-level correlations between query-memory frame enjoy higher weights. Besides, compared to the baseline STCN, we achieve better consistent segmentation results for object *skateboard* even with drastic appearance variations caused by the movement of objects and cameras (*i.e.*, *4bef684040* in Figure 2), credited to modeling the temporal trajectory in a data-driven manner. Please refer to attached *qualitative_for_TMRN*.mp4 for the qualitative video.

References

[1] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: toward class-agnostic and very high-resolution segmentation via global and local refinement. In *CVPR*, pages 8890–8899, 2020. 2

[2] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, pages 640–658. Springer, 2022. 1, 2

[3] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *NIPS*, 2021. 1, 2, 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[6] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *CVPR*, pages 2869–2878, 2020. 2

[7] 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 *ECCV*, pages 648–665. Springer, 2022. 2, 3

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[9] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9226–9235, 2019. 1, 2, 3

[10] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 2

[11] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *PAMI*, 38(4):717–729, 2015. 2

[12] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017. 2

[13] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos:

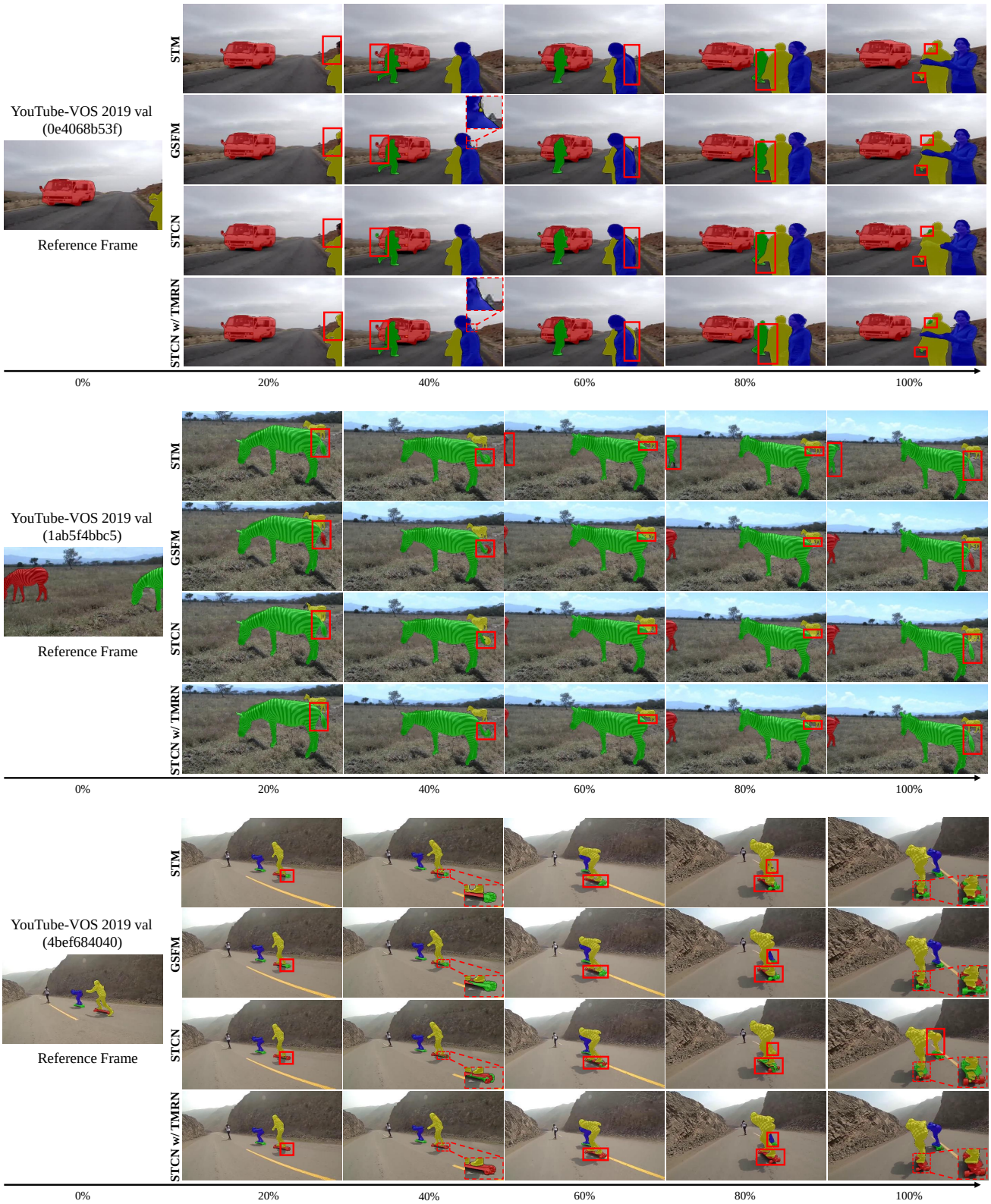

Figure 2. More qualitative results on YouTube-VOS 2019 validation set. We mark significant improvements from STM [9], GSFM [7] and STCN [3] using red boxes.

A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 2

[14] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *ICCV*, pages 7234–7243, 2019. 2