XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model

Ho Kei Cheng and Alexander G. Schwing

Abstract

We present XMem, a video object segmentation architecture for long videos with unified feature memory stores inspired by the Atkinson-Shiffrin memory model. Prior work on video object segmentation typically only uses one type of feature memory. For videos longer than a minute, a single feature memory model tightly links memory consumption and accuracy. In contrast, following the Atkinson-Shiffrin model, we develop an architecture that incorporates multiple independent yet deeply-connected feature memory stores: a rapidly updated sensory memory, a high-resolution working memory, and a compact thus sustained long-term memory. Crucially, we develop a memory potentiation algorithm that routinely consolidates actively used working memory elements into the long-term memory, which avoids memory explosion and minimizes performance decay for long-term prediction. Combined with a new memory reading mechanism, XMem greatly exceeds state-of-the-art performance on long-video datasets while being on par with state-of-the-art methods (that do not work on long videos) on short-video datasets.

Keywords： video object segmentation, feature memory stores, memory potentiation algorithm.

1 Introduction

This paper focuses on semi-supervised video object segmentation. Authors frame Video Object Segmentation (VOS), first and foremost, as a memory problem. By studying prior work on VOS, authors of the paper found existent models consume large amount of memory, and a single feature memory model tightly links memory consumption and accuracy in long videos.

Inspired by the Atkinson-Shiffrin memory model, authors present XMem, a video object segmentation architecture for long videos with unified feature memory stores, which include a rapidly updated sensory memory, a high-resolution working memory, and a compact thus sustained long-term memory. They are equipped with a memory reading operation for high-quality video object segmentation on both long and short videos.

Besides, authors develop a memory consolidation algorithm that selects representative prototypes from the working memory, and a memory potentiation algorithm that enriches these prototypes into a compact yet powerful representation for long-term memory storage.

2 Related works

2.1 General VOS Methods

The authors list many common methods with their defects, including those who employ a feature memory to store information given in the first frame, online learning approaches and recent improvements, tracking-based approaches, and recent state-of-the-art methods that use more past frames as feature memory. Amongst

them, STM, STCN, and AOT are particularly mentioned to conquer defects of prior methods but still remain other defects. In sum, prior works mostly use a single type of feature memory.

At last, authors propose their XMem that can keep the GPU memory usage strictly bounded due to their long-term memory and consolidation.

2.2 Methods that Specialize in Handling Long Videos

AFB-URR proposed by Liang et al. and global context module proposed by Li et al. both eagerly compress new high-resolution feature memory into a compact representation, thus sacrificing segmentation accuracy, while authors' multi-store feature memory avoids eager compression and achieves much higher accuracy in both short-term and long-term predictions.

3 Method

3.1 Overview

Authors describe the process of XMem that combines sensory memory, working memory and long-term memory together to provide high-quality features with low GPU memory usage even for very long videos.

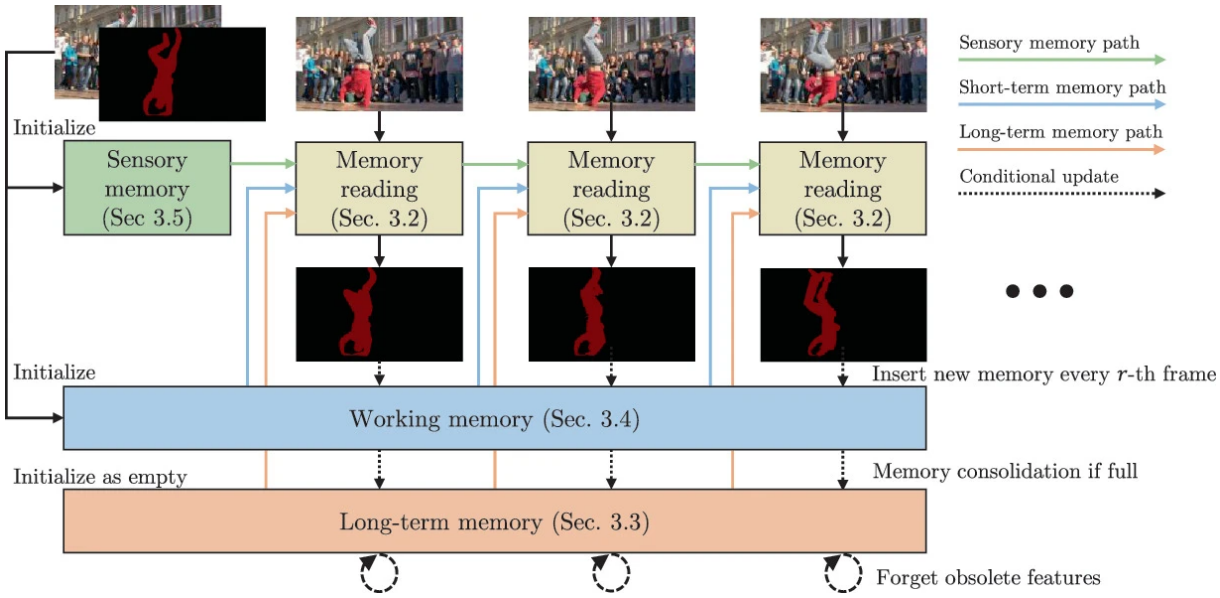


Figure 1: Overview of XMem

Figure 1 provides an overview of XMem. Given the image and target object mask at the first frame (top-left of Fig. 1), XMem tracks the object and generates corresponding masks for subsequent query frames. For this, XMem first initialize the different feature memory stores using the inputs. For each subsequent query frame, XMem perform memory reading from long-term memory, working memory, and sensory memory respectively. The readout features are used to generate a segmentation mask. Then, XMem update each of the feature memory stores at different frequencies. XMem update the sensory memory every frame and insert features into the working memory at every r-th frame. When the working memory reaches a pre-defined maximum of T_{max} frames, XMem consolidate features from the working memory into the long-term memory in a highly compact form. When the long-term memory is also full, XMem discard obsolete features to bound the maximum GPU memory usage.

3.2 Memory Reading

Authors illustrate the process of memory reading and mask generation for a single frame. The feature F representing information stored in both the long-term and the working memory is computed via the readout operation

$$F = vW(k, q)$$

, where k and v are C^k- and C^v- dimensional keys and values for a total of N memory elements which are stored in both the long-term and working memory. And the q is query gained through a feature extractor ResNet-50.

3.3 Long-Term Memory

Authors discuss motivation, prototype selection, memory potentiation, and removing obsolete features in long-term memory stores. The mechanism aims to enable efficient and accurate segmentation of long videos. The long-term select a small representative subset from the candidates as prototypes, and store them in compact form. When a pre-defined memory limit is reached, the long-term memory discards obsolete features to keep the scale of memory.

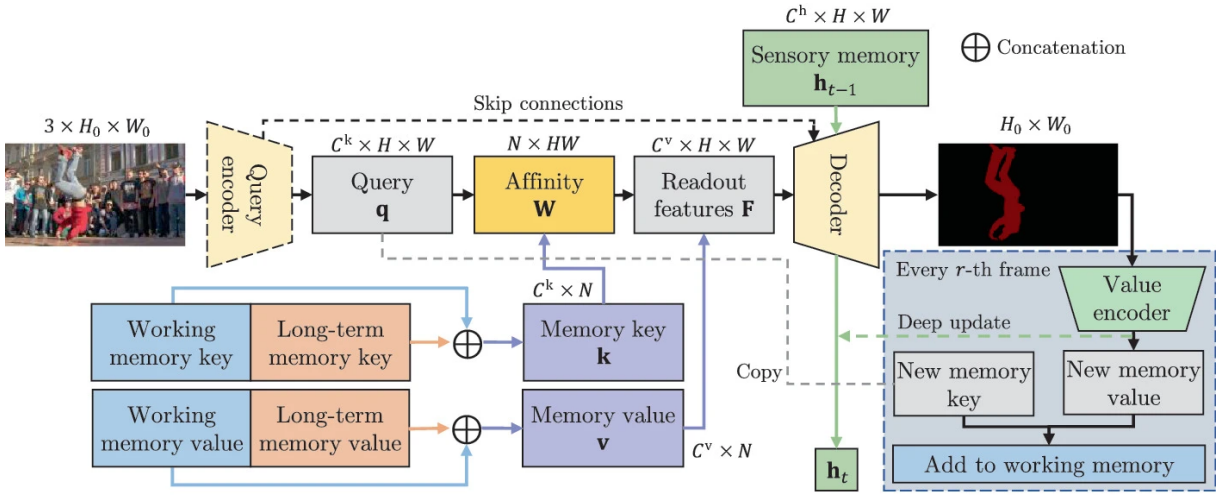


Figure 2: Process of memory reading and mask decoding of a single query frame.

3.4 Working Memory

Working memory is crucial for accurate short-term prediction. It acts as the basis for the long-term memory. The working memory consists of keys and values. The key is encoded from the image and resides in the same embedding space as the query q while the value is encoded from both the image and the mask. The working memory updates every r-th frame. Bottom-right of Fig. 2 illustrates the working memory update process.

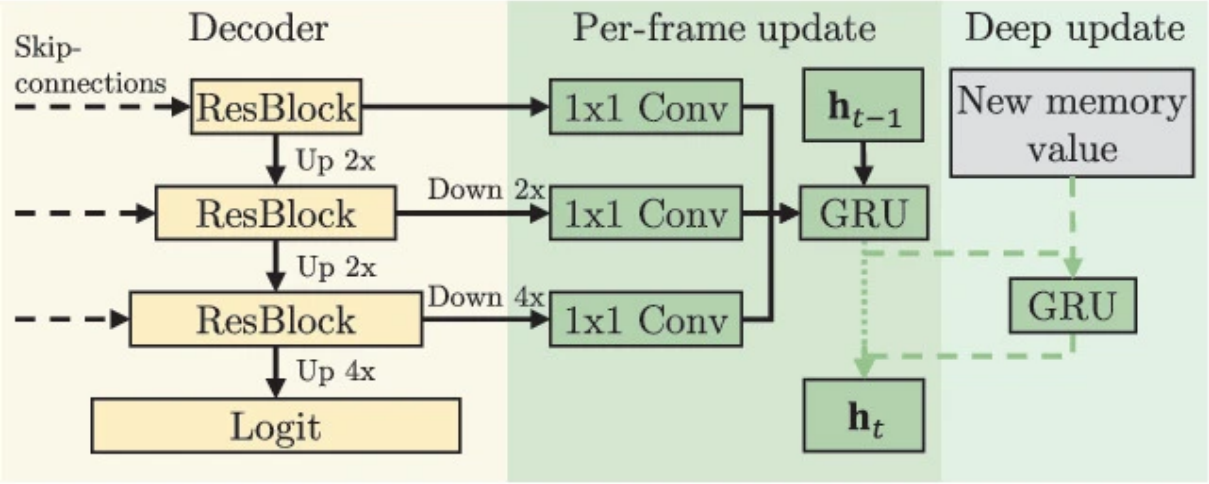


Figure 3: Sensory memory update overview.

3.5 Sensory Memory

The sensory memory focuses on the short-term and retains low-level information. It updates every frame, discarding redundant information that has already been saved to the working memory, and receiving updates from a deep network with minimal overhead. The process is shown in Fig. 1, where GRU is short for "Gated Recurrent Unit".

4 Implementation Details

4.1 Networks

The authors adopt ResNets as the feature extractor, removing the classification head and the last convolutional stage. To generate the query q, the shrinkage term s, and the selection term e, the authors apply separate 3×3 convolutional projections to the query encoder feature output. The authors set $C^k = 64$, $C^v = 512$, and $C^h = 64$. To control the range of the shrinkage factor to be in [1,∞), the authors apply $(\cdot)^2 + 1$, and to control the range of the selection factor to be in [0, 1], the authors apply a sigmoid. In the multi-object scenario, the authors use soft-aggregation to fuse the final logits from different objects.

4.2 Training

The authors first pretrain the network on synthetic sequences of length three generated by deforming static images. They adopt the open-source implementation of STCN without modification. Next, the authors perform the main training on YouTubeVOS and DAVIS with curriculum sampling. Noting that the default sequence length of three is insufficient to train the sensory memory as it would be heavily dependent on the initial state, the authors instead sample sequences of length eight. To reduce training time and for regularization, a maximum of three (instead of all) past frames are randomly selected to be the working memory for any query in training time.

The authors use bootstrapped cross entropy loss and dice loss with equal weighting. For optimization, the authors use AdamW with a learning rate of 1e-5 and a weight decay of 0.05, for 150K iterations with batch size 16 in static image pretraining, and for 110K iterations with batch size 8 in main training. The authors drop the learning rate by a factor of 10 after the first 80K iterations. For a fair comparison, the authors also retrain

the STCN baseline with the above setting. There is no significant difference in performance for STCN.

4.3 Experimental Environment Setup

My environment is as follows.

NVIDIA-SMI 460.32.03; Driver Version: 460.32.03; CUDA Version: 11.2; GPU: Tesla T4.

4.4 Steps

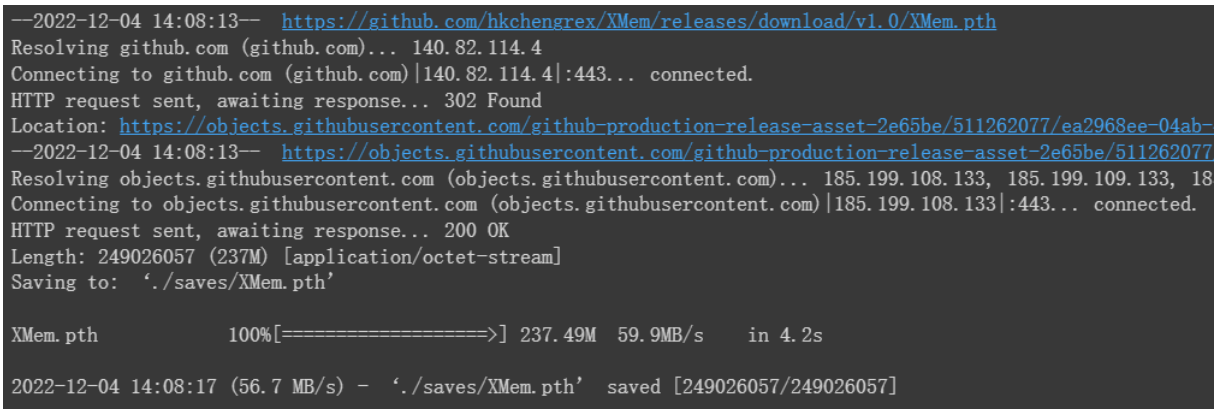


Figure 4: Download the pretrained model.

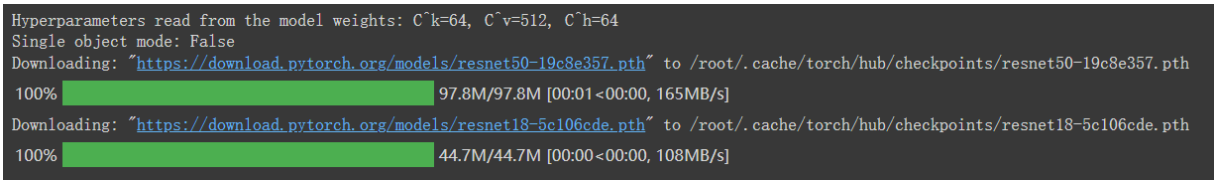


Figure 5: Perform basic setup.

Figure 6: A long sample video.

Firstly, I get the code and install prerequisites. To train the model, I use the provided 'scripts/download_datasets.py' to structure the datasets as required format. The datasets contain long time videos such as extended version and corresponding '_davis' versions of AFB-URR. I use 'eval.py' to execute command line inference. In testing, I download the pretrained model (in Fig. 4) and perform basic setup (in Fig. 5). I load some data from the source of Youtube. After previewing the video and first-frame annotation as shown in Fig.

6 and Fig. 7, I convert the mask to a numpy array, where the object IDs should be consecutive and start from 1 (0 represents the background). Last I propagte masks frame-by-frame, getting segmentation results from a long video.

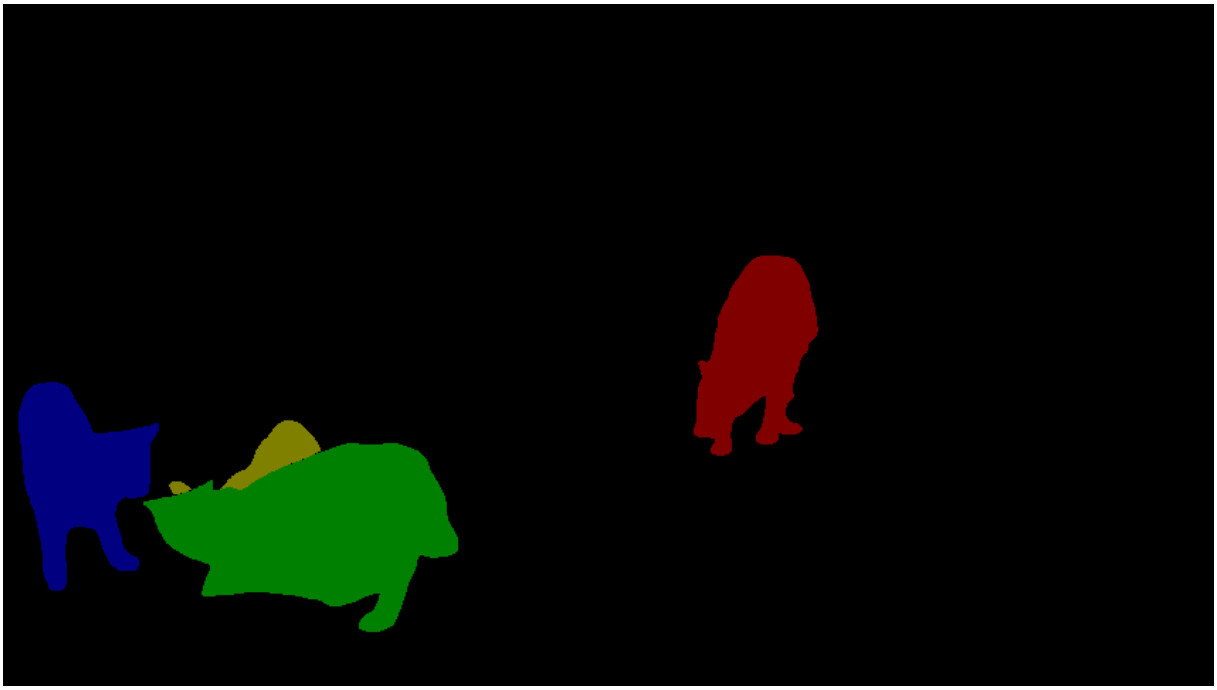


Figure 7: The first frame of the sample video.

The official code is available at https://github.com/hkchengrex/XMem.

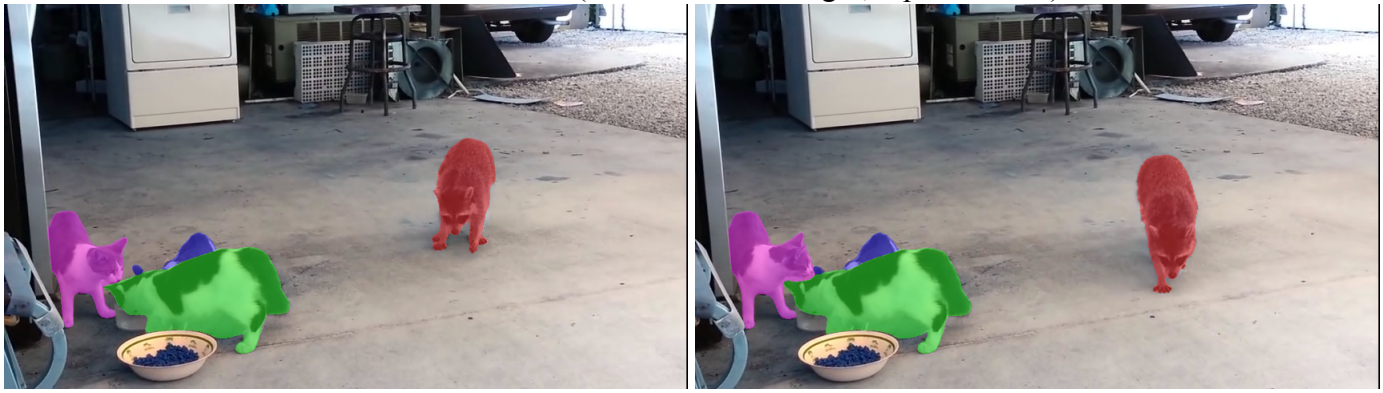
4.5 Main contributions

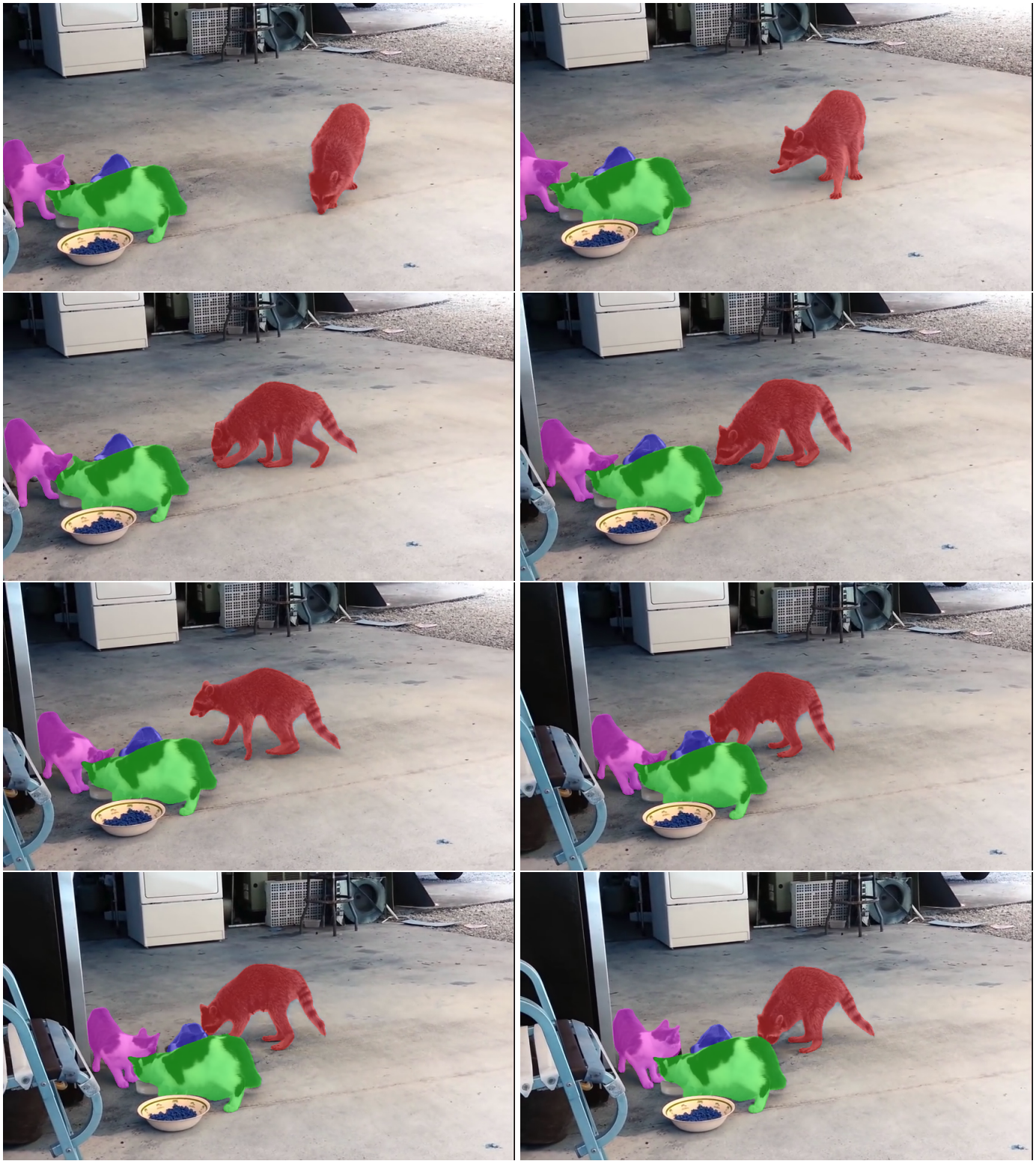
Methods with a short memory span are not robust to changes, while those with a large memory bank are subject to a catastrophic increase in computation and GPU memory usage. Attempts at long-term attentional VOS like AFB-URR compress features eagerly as soon as they are generated, leading to a loss of feature resolution.

The authors' method is inspired by the Atkinson-Shiffrin human memory model, which has a sensory memory, a working memory, and a long-term memory. These memory stores have different temporal scales and complement each other in the memory reading mechanism. It performs well in both short-term and long-term video datasets, handling videos with more than 10,000 frames with ease.

5 Results and analysis

The result frames are shown as follows (read from left to right, top to bottom).





The objects of interest (cats and a raccoon) in the video are precisely segmented and tracked all along. The entire training process takes around 35 h on two RTX A6000 GPUs. Deep updates are performed with a probability of 0.2. And the segmentation job for the video of 45 seconds takes only about 1 minute and cost about 0.6 GB memory.

6 Conclusion and future work

6.1 Conclusion

XMem performs accurate segmentation with minimal GPU memory usage for both long and short videos with more than 10,000 frames with ease, only requiring user to provide the first frame. It outperforms SOTA in long videos with being on par with other SOTA in short videos.

6.2 Future Work

The feature of few GPU memory usage enable XMem to be used in mobile devices for VOS easily. However, the method sometimes fails when the target object moves too quickly or has severe motion blur as even the fastest updating sensory memory cannot catch up, such as flying birds, a thrown frisbee, and waving flags. This defect enables improving work meaningful.

参考文献

[1] Atkinson, R.C., Shiffrin, R.M.: Human memory: a proposed system and its control processes. In: Psychology of learning and motivation, vol. 2, pp. 89–195. Elsevier (1968)

[2] Bhat, G., et al.: Learning what to learn for video object segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 777 – 794. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_46

[3] Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR (2017)

[4] Chang, A.X., et al.: ShapeNet: an information-rich 3D model repository. arXiv:1512.03012 (2015)

[5] Chen, X., Li, Z., Yuan, Y., Yu, G., Shen, J., Qi, D.: State-aware tracker for real-time video object segmentation. In: CVPR (2020)

[6] Chen, Y., Pont-Tuset, J., Montes, A., Van Gool, L.: Blazingly fast video object segmentation with pixel-wise metric learning. In: CVPR (2018)

[7] Cheng, H.K., Chung, J., Tai, Y.W., Tang, C.K.: Cascadepsp: toward class-agnostic and very high-resolution segmentation via global and local refinement. In: CVPR (2020)

[8] Cheng, H.K., Tai, Y.W., Tang, C.K.: Modular interactive video object segmentation: interaction-to-mask, propagation and difference-aware fusion. In: CVPR (2021)

[9] Cheng, H.K., Tai, Y.W., Tang, C.K.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In: NeurIPS (2021)

[10] 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 (2018)

[11] Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. arXiv (2014)

[12] Denninger, M., et al.: Blenderproc. arXiv:1911.01911 (2019)

[13] Duarte, K., Rawat, Y.S., Shah, M.: Capsulevos: semi-supervised video object segmentation using capsule routing. In: ICCV (2019)

[14] Duke, B., Ahmed, A., Wolf, C., Aarabi, P., Taylor, G.W.: Sstvos: sparse spatiotemporal transformers for video object segmentation. In: CVPR (2021)

[15] Forsyth, D., Ponce, J.: Computer Vision: A Modern Approach. Prentice hall, Upper Saddle River (2011)

[16] Ge, W., Lu, X., Shen, J.: Video object segmentation using global and instance embedding learning. In: CVPR (2021)

[17] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

[18] Hu, L., Zhang, P., Zhang, B., Pan, P., Xu, Y., Jin, R.: Learning position and target consistency for memory-based video object segmentation. In: CVPR (2021)

[19] Hu, P., Wang, G., Kong, X., Kuen, J., Tan, Y.P.: Motion-guided cascaded refinement network for video object segmentation. In: CVPR (2018)

[20] Hu, Y.T., Huang, J.B., Schwing, A.: Maskrnn: instance level video object segmentation. In: NIPS (2017)

[21] Huang, X., Xu, J., Tai, Y.W., Tang, C.K.: Fast video object segmentation with temporal aggregation network and dynamic template matching. In: CVPR (2020)

[22] Jang, W.D., Kim, C.S.: Online video object segmentation via convolutional trident network. In: CVPR (2017)

[23] Johnander, J., Danelljan, M., Brissman, E., Khan, F.S., Felsberg, M.: A generative appearance model for end-to-end video object segmentation. In: CVPR (2019)

[24] Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)

[25] Li, X., Wei, T., Chen, Y.P., Tai, Y.W., Tang, C.K.: Fss-1000: a 1000-class dataset for few-shot segmentation. In: CVPR (2020)

[26] Li, X., Loy, C.C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 93–110. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_6

[27] Li, Yu., Shen, Z., Shan, Y.: Fast video object segmentation using the global context module. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12355, pp. 735–750. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58607-2-43

[28] Liang, Y., Li, X., Jafari, N., Chen, J.: Video object segmentation with adaptive feature bank and uncertain-region refinement. In: NeurIPS (2020)

[29] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)

[30] Lu, X., Wang, W., Danelljan, M., Zhou, T., Shen, J., Van Gool, L.: Video object segmentation with episodic graph memory networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12348, pp. 661–679. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58580-8-39

[31] Maninis, K.K., et al.: Video object segmentation without temporal information. PAMI 41, 1515–1530 (2018)

[32] Mao, Y., Wang, N., Zhou, W., Li, H.: Joint inductive and transductive learning for video object segmentation. In: ICCV (2021)

[33] Meinhardt, T., Leal-Taixé, L.: Make one-shot video object segmentation efficient again. In: NeurIPS (2020)

[34] Oh, S.W., Lee, J.Y., Sunkavalli, K., Kim, S.J.: Fast video object segmentation by reference-guided mask propagation. In: CVPR (2018)

[35] Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019)

[36] Park, H., Yoo, J., Jeong, S., Venkatesh, G., Kwak, N.: Learning dynamic network using a reuse gate function in semi-supervised video object segmentation. In: CVPR (2021)

[37] Patrick, M., et al.: Keeping your eye on the ball: trajectory attention in video transformers. In: NeurIPS (2021)

[38] Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR (2017)

[39] 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 (2016)

[40] Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv:1704.00675 (2017)

[41] Robinson, A., Lawin, F.J., Danelljan, M., Khan, F.S., Felsberg, M.: Learning fast and robust target models for video object segmentation. In: CVPR (2020)

[42] Seong, H., Hyun, J., Kim, E.: Kernelized memory network for video object segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12367, pp. 629–645. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58542-6-38

[43] Seong, H., Oh, S.W., Lee, J.Y., Lee, S., Lee, S., Kim, E.: Hierarchical memory matching network for video object segmentation. In: ICCV (2021)

[44] Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended cssd. TPAMI 38, 717–729 (2015)

[45] Squire, L.R., Genzel, L., Wixted, J.T., Morris, R.G.: Memory consolidation. In: Cold Spring Harbor perspectives in biology. Cold Spring Harbor Lab (2015)

[46] 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 (2019)

[47] 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 (2019)

[48] Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017)

[49] Wang, H., Jiang, X., Ren, H., Hu, Y., Bai, S.: Swiftnet: real-time video object segmentation. In: CVPR (2021)

[50] Wang, L., et al.: Learning to detect salient objects with image-level supervision. In: CVPR (2017)

[51] Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: a unifying approach. In: CVPR (2019)

[52] Wang, Z., Xu, J., Liu, L., Zhu, F., Shao, L.: Ranet: ranking attention network for fast video object segmentation. In: ICCV (2019)

[53] Xie, H., Yao, H., Zhou, S., Zhang, S., Sun, W.: Efficient regional memory network for video object segmentation. In: CVPR (2021)

[54] Xiong, Y., et al.: Nyströmformer: A nyström-based algorithm for approximating self-attention. In: AAAI (2021)

[55] Xu, K., Wen, L., Li, G., Bo, L., Huang, Q.: Spatiotemporal CNN for video object segmentation. In: CVPR (2019)

[56] Xu, N., et al.: Youtube-vos: a large-scale video object segmentation benchmark. In: ECCV (2018)

[57] Xu, X., Wang, J., Li, X., Lu, Y.: Reliable propagation-correction modulation for video object segmentation. In: AAAI (2022)

[58] Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by foreground-background integration. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12350, pp. 332–348. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7-20

[59] Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. In: NeurIPS (2021)

[60] Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by multi-scale foreground-background integration. PAMI (2021)

[61] Zeng, Y., Zhang, P., Zhang, J., Lin, Z., Lu, H.: Towards high-resolution salient object detection. In: ICCV (2019)

[62] Zhang, L., Lin, Z., Zhang, J., Lu, H., He, Y.: Fast video object segmentation via dynamic targeting network. In: ICCV (2019)

[63] Zhang, Y., Wu, Z., Peng, H., Lin, S.: A transductive approach for video object segmentation. In: CVPR (2020)