2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

SAM2-LongVideo: a robust tracking system based on SAM2 and YOLO

Saijun Wang[1][0009-0009-2528-1914]

[1] School of Electronics and Information Engineering, Tongji University, Shanghai, China
2151944@tongji.edu.cn

Abstract. Video Object Segmentation (VOS) aims to track and segment objects across video sequences at the pixel level. SAM2, a state-of-the-art segmentation model, excels in zero-shot image and video segmentation but faces challenges in long video processing, including object tracking loss and high memory usage. This paper proposes a solution by combining SAM2 with YOLOv11. YOLOv11 provides real-time bounding box prompts for efficient segmentation, while SAM2 segments the target across sub-sequences. This approach reduces memory requirements and prevents object tracking loss by dividing the video into smaller sub-sequences without laborious human annotation. Experimental results show that the method maintains high segmentation accuracy with minimal memory consumption, making it suitable for long video segmentation in resource-constrained environments. This method offers an efficient and scalable solution for VOS tasks in various applications. The codes and GUI are available at https://github.com/Saijun-Wang/SAM2-LongVideo.

Keywords: Video Object Segmentation, SAM2, YOLOv11, Long Video Segmentation, Memory Optimization.

1 Introduction

1.1 Video Object Segmentation (VOS)

VOS is a classic task in computer vision that aims to consistently track the same object or multiple targets over a video sequence and perform pixel-level segmentation. This task has deep practical applications in various fields such as autonomous driving, medical imaging, remote sensing, video editing, and robot control. The task's complexity arises from the need to consistently identify and separate objects in dynamic and often unpredictable video environments.

Currently, deep learning methods have played a significant role in improving the performance of VOS across various applications, enabling better segmentation accuracy and the handling of more complex scenarios. However, before the emergence of large language models (LLMs), VOS typically required the design of specialized segmentation models for different application scenarios, making it difficult for a general model to be applied across diverse videos or tasks. This lack of generalization has posed significant limitations, as many models were not designed with transfer learning capabilities in mind, making them unable to adapt to new, unseen data—thus struggling with real-world video content that changes rapidly over time.

In addition, most of these traditional approaches have been computationally intensive, requiring substantial amounts of memory and processing power, which can be a bottleneck when working with long video sequences. While recent advancements in VOS have shown promise, challenges such as object tracking loss and high memory consumption in long video processing remain significant issues. These obstacles hinder the application of VOS in real-world scenarios, especially in resource-constrained environments.

1.2 Segment Anything Model 2 (SAM2)

The emergence of large scale vision models such as SAM2[1], has made notable strides in overcoming these barriers. SAM2 requires user interaction through the provision of points, bounding boxes (bboxes), or masks as prompts. These prompts are then propagated across the entire video stream to achieve segmentation. Due to training on large datasets, SAM2 has achieved excellent zero-shot performance and has become a vision foundational model, offering significant improvements in flexibility and adaptability to new tasks. However, SAM2 faces challenges when it comes to processing long videos.

Firstly, if the prompt is provided on the first frame, SAM2 tends to lose track of the target after several dozen to a few hundred frames, especially in the presence of occlusions or similar objects. This occurs because SAM2 stores the mask of the prompt frame and the memory of the most recent frames in its memory bank, meaning its inference is based on information from adjacent frames. As a result, if the target object does not appear in the recent frames and undergoes some deformation compared to its appearance in the first frame, SAM2 struggles to segment the object correctly in the current frame. In fact, this happens frequently.

Secondly, SAM2 uses a streaming memory approach for the image encoder, which requires significant memory storage and segmentation time. For example, processing 1,000 frames of 1280x720 images demands around 17GB of GPU memory. Given that 1,000 frames correspond to only 33 seconds of video at 30 frames per second (FPS), this is not even considered a long video. In scenarios with limited computational resources, segmenting a short video can still require substantial time. For instance, on an RTX 2080 Ti (22GB), segmenting 1,000 frames takes about 7 minutes. Therefore, SAM2 has high computational resource requirements and is not suitable for embedded devices or environments with limited personal computing resources.

In summary, due to these two limitations, long videos must be split into multiple segments for processing, which requires multiple manual prompts, making the process extremely laborious.

After the release of SAM2, many studies have attempted to improve it. However, most research papers on SAM2 focus on improving segmentation accuracy in general or adapting it to specific downstream tasks, with few paying attention to these two challenges, though achieving accurate segmentation with low computational cost is crucial in real-world deployments. In resource-constrained settings, those two issues mentioned above pose significant obstacles to users.

Currently, very few studies have paid attention to them, and none solved them perfectly. Liu et al. [2] focused on high-resolution medical surgery videos that last for several hours, selectively reducing redundant frames with less informative content. However, their core idea of ignoring less important frames cannot be widely transferred to other non-medical surgery domains, as not all scenarios allow for the omission of certain frames, making this approach unsuitable for general-purpose models.

1.3 You Only Look Once

You Only Look Once (YOLO) [3] is a popular real-time object detection model. The main advantage of YOLO is its efficiency, as it can detect objects in a very short amount of time, outputting the object classes, bboxes, and confidence scores. The average detection speed of YOLO is around 20 ms per image.

YOLOv11 [4] is the latest version of YOLO, offering different modes such as detect, segment, and classify. However, YOLO has several limitations. For instance, the segmentation accuracy of YOLOv11-seg is far behind that of SAM, and its object predictions are based on the complete object contours, without treating multiple parts or incomplete regions as a single object, as some semantic segmentation models do. In real-world scenarios, many objects are subject to occlusion and consist of multiple disconnected parts, and in such cases, YOLO fails to segment the object properly. Additionally, YOLO is a fully supervised model that requires a large number of labeled images for training. Therefore, obtaining these annotations is a necessary and time-consuming task.

Hence this paper proposes a comprehensive approach for VOS by combining the YOLOv11 model with the SAM2 to develop a robust tracking system without laborious annotation. This method first divides the image sequence into different sub-sequences, and within each sub-sequence, it utilizes YOLO11's detection head to quickly and real-time provide bbox prompts, thereby achieving efficient and accurate segmentation of long videos. Moreover, this approach can also be easily adapted to other higher-quality variants of SAM2, providing future researchers with a more straightforward and more efficient solution for long video segmentation.

To address these challenges, this paper proposes a novel approach that combines SAM2 with YOLOv11, an advanced object detection model. In this method, YOLOv11 provides real-time bounding box prompts that facilitate efficient segmentation, while SAM2 focuses on segmenting the object across smaller, more manageable sub-sequences. This method allows for a reduction in memory consumption and mitigates tracking loss by dividing the video into smaller sub-sequences, without requiring extensive human annotation.

The experimental results demonstrate that this approach successfully maintains high segmentation accuracy while minimizing memory usage, making it ideal for long video segmentation tasks in environments with limited resources. This hybrid method offers a scalable and efficient solution for VOS in a variety of practical applications.

2 Method

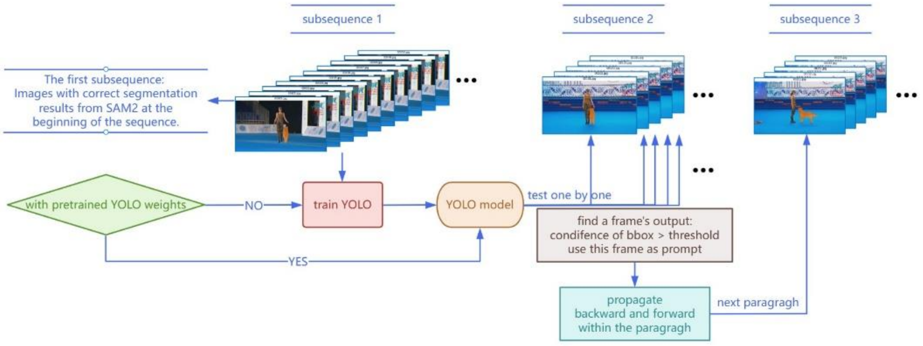

Fig. 1. model structure

SAM2-LongVideo combines the advantages of SAM2 and YOLO to address two common challenges in long VOS: object tracking loss due to occlusion and high memory consumption. In long videos, especially when an object disappears for a period of time, SAM2 tends to lose track of the object. Besides, after the object is occluded, it may also start tracking other similar objects by mistake. In this approach, YOLO's detection head automatically provides prompts, while SAM2 is responsible for segmentation and tracking.

2.1 Assumption

For long videos, objects to be tracked are often subject to occlusion. In this method, it is assumed that the target object remains visible in the frame for a continuous sequence of 200–300 frames with minimal occlusion. Additionally, it is assumed that there are no objects in the scene that are highly similar to the target object.

2.2 Mechanism

First the video frame sequence is divided into multiple sub-sequences. The length of each sub-sequence can be specified by the user based on the task requirements. For fast-moving objects that are frequently occluded, a recommended sub-sequence length is 50–100 frames. For slow-moving objects that are occasionally occluded, a length of 100–150 frames is more suitable. Since SAM2-LongVideo segments only one sub-sequence at a time, the GPU memory it requires is equivalent to storing just 50–150 frames, which makes it feasible to run on GPUs with as little as 4GB of memory.

For users, it is only necessary to provide the coordinates of the object to be tracked and segmented in the first frame. This mechanism is consistent with SAM2, allowing users to add several positive and negative points for a single object. The information from the prompt frame is stored in SAM2's memory module, and SAM2 propagates this information within the first sub-sequence to segment the target object. Typically, the first sub-sequence should contain at least 150 frames, which is longer than the subsequent sub-sequences.

The segmentation results from the first sub-sequence are then used to generate training dataset for training the YOLO detection head. Although this training process requires significant GPU memory, it only needs to be done once for the same object across multiple long videos, with the trained model weights being reusable. The trained YOLO detection head is responsible for providing the target object's position in subsequent sub-sequences. In practice, each sub-sequence only requires one accurate prompt frame (which does not have to be the first frame), and SAM2 can propagate this prompt both forward and backward to segment the entire sub-sequence. Therefore, the YOLO detection head does not need to be perfectly trained. In experiments, training YOLOv11s on 150 images for about 300 epochs is sufficient. Additionally, to further reduce GPU memory usage during YOLO training, the resolution of training images can be lowered. However, experiments show that models trained on lower-resolution images experience a decrease in confidence when detecting objects in original-resolution images. If the model is trained on images resized to 512×512 and the trained YOLO are then used to detect objects in both resized 512×512 images and original 1280×720 images, the detection results are shown in Fig.2. It can be observed that the detection confidence is higher for the resized images on the left.

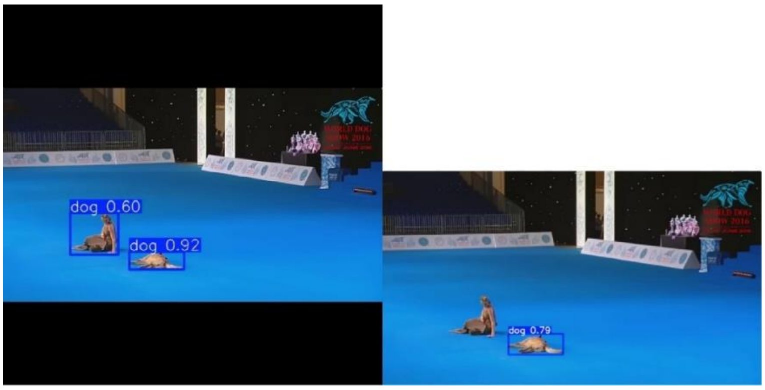

Fig. 2. YOLO detection result(left image is resized to 512×512, right image is 1280×720) [5]

Within each sub-sequence, due to YOLO's fast and real-time detection capabilities, it can quickly identify a frame with confidence above a certain threshold to serve as the prompt. In this case, the prompt provided by the YOLO detection head with high confidence is considered reliable.

In long video sequences, providing a point prompt only in the first frame often leads to tracking failure as the temporal distance from the prompt increases. Moreover, once a segmentation error occurs in a frame, subsequent frames are likely to continue this error due to the memory of the incorrect frame being retained. A common solution in SAM2 is to provide additional point prompts at error frames to correct the memory and re-segment. However, this correction process requires manual supervision to monitor segmentation performance and timely addition of prompts, which is time-consuming and labor-intensive. To address this issue, SAM2-LongVideo divides long sequences into shorter sub-sequences, enabling independent prompt propagation within each sub-sequence. Since the long video is split into shorter parts, each frame remains close to the prompt frame, reducing the likelihood of losing track. Even if tracking is lost in one sub-sequence, it only affects that particular sub-sequence without impacting the previous or next ones. Whether a sub-sequence loses track depends solely on the accuracy of the YOLO detection head. The sub-sequences are independent of each other, which actually means that they can be computed in parallel as well.

Additionally, according to the study by Zhou et al. [6] sing bboxes and masks as prompts is more precise than using points. In SAM2-LongVideo, the YOLO detection head adopts bbox prompts. There are two reasons. First, the YOLO11-seg training dataset labels objects with contours rather than the mask format used in SAM2, making it difficult to handle objects that appear as multiple disconnected regions due to occlusion. Second, YOLO's segmentation mode has relatively low accuracy, even with the pretrained weights. Thus the bbox prompt is adopted.

3 Results

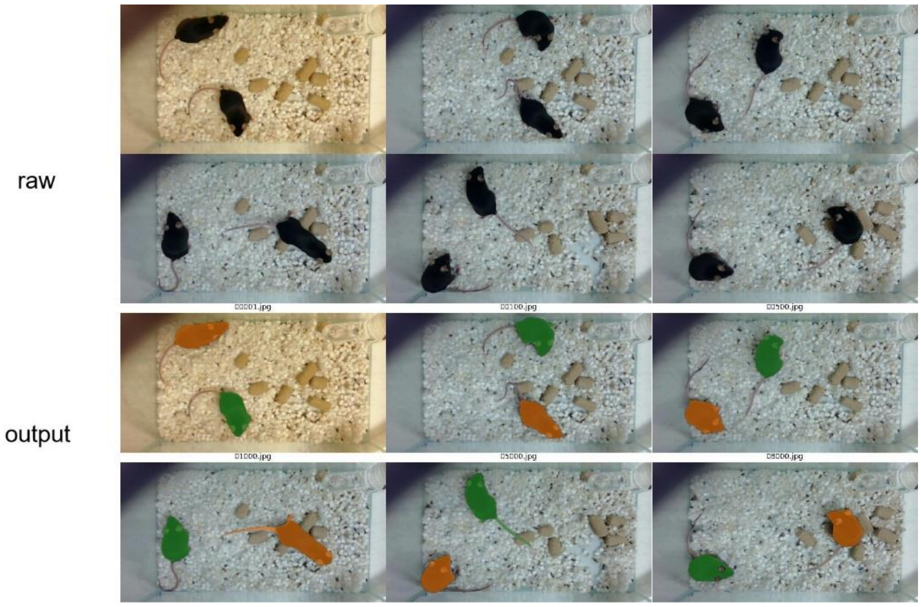

Fig. 3. Segmentation of a mice video

Fig.3 shows an example of segmenting 8000 frames of mice video images with this method. If using SAM2 for segmentation, it is not possible to segment such a large number of images at once on a single GPU. However, by employing the SAM2-LongVideo method, all segmentation results can be obtained at one time without losing track.

The camera is positioned directly above the mouse incubator. By extracting the pixel masks of the mice, corresponding bboxes are generated. The center of each bbox is considered the position of the mouse. Then the distance between the two mice is computed throughout the video, resulting in Figure 4, which shows the distances for frames 1-3000, 3001-6000, and 6001-8979, respectively.

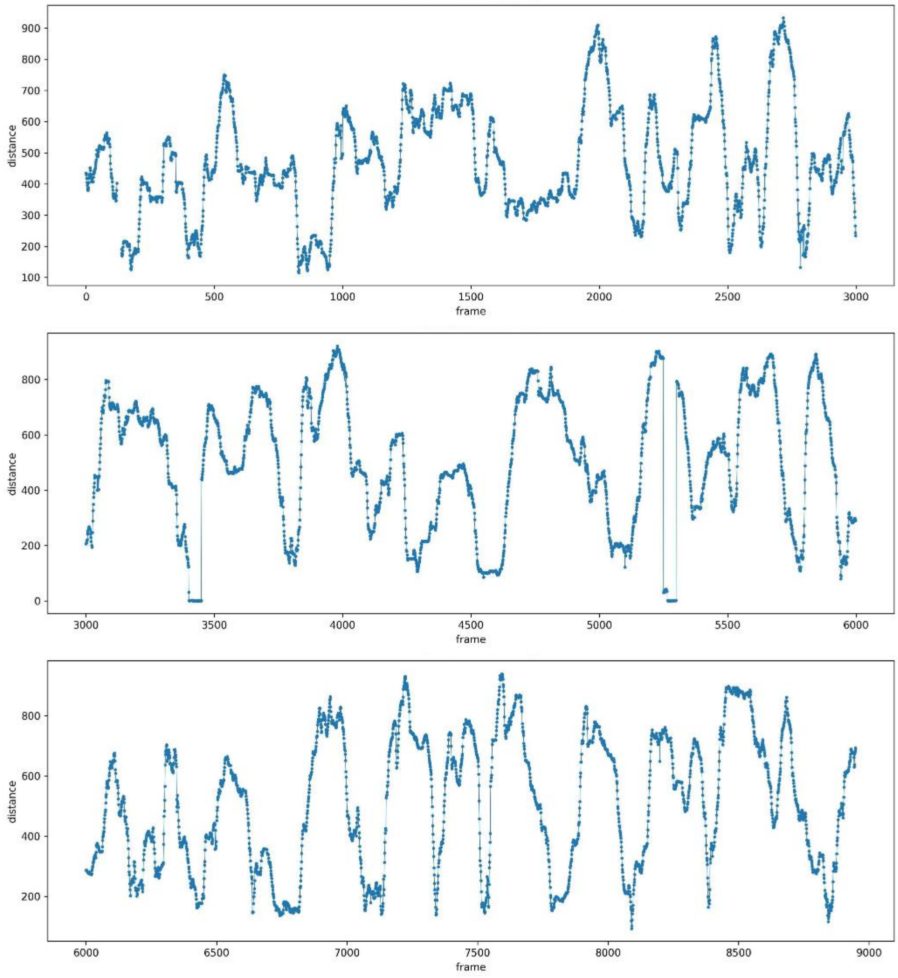

Fig. 4. Distance Between Two Mice (Frames 1-8979)

From the three result graphs, it can be observed that the distance between the mice changes continuously over time. This means that the method successfully tracked both mice. However, there were indeed two instances where the distance suddenly dropped to zero, around frame 3400 and frame 5200. In these frames, the method failed to track both mice simultaneously and only detected one. Nevertheless, such occurrences were rare.

Besides, there is also a sudden disappearance in the measured distance around frame 130. This is due to temporary blackouts caused by camera occlusion. Overall, the SAM2-LongVideo method maintains consistent tracking throughout the long video sequence, demonstrating its strong tracking capability. This also suggests that the method is well-suited for animal behavior tracking and analysis.

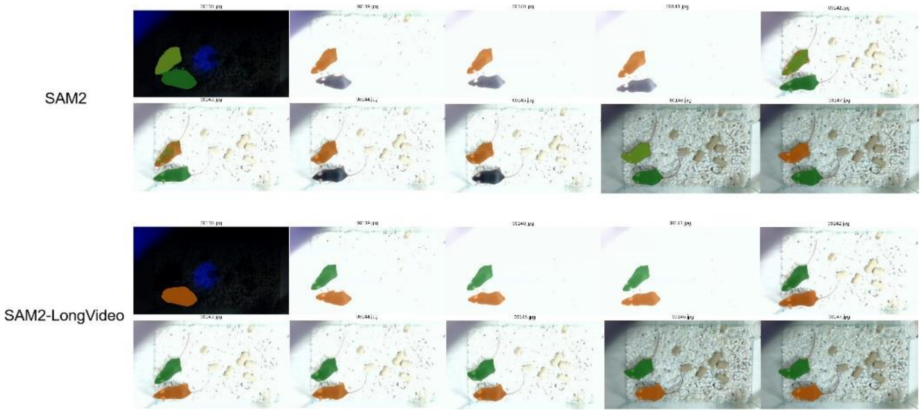

Fig. 5. Comparison results between SAM2 and SAM2-LongVideo after occlusion

Moreover, SAM2 often struggles with segmentation errors in frames where an object disappears and then reappears. For instance, as shown in Figure 5, SAM2 produces incorrect segmentations in the several frames immediately after the blackout ends and the scene becomes visible again. Although these errors are corrected in the following frames in this particular example, SAM2 does not always manage to rectify them successfully.

In contrast, SAM2-LongVideo avoids this issue in this case by utilizing a high-confidence frame as a prompt, allowing the memory stream to propagate information from a later frame back to the preceding frame, rather than solely relying on forward propagation from the initial frame.

Moreover, calculating the distance between mice primarily relies on generating a bbox based on the center of their body parts (excluding the tail) and then computing the Euclidean distance between the bbox centers. Therefore, it is crucial to avoid segmenting the mice’s tails. Although I applied a negative prompt for each mouse’s tails in the initial frame, SAM2 still segmented them in many subsequent frames. However, under the same initial prompt, SAM2-LongVideo, which utilizes the bbox provided by the YOLO detection head as a prompt, improves the segmentation of the main body and its connected parts. While it still segments the tails in some frames, the overall occurrence is significantly reduced compared to pure SAM2 (168 out of 2000 frames for SAM2, whereas SAM2-LongVideo only has 57 out of 1,000 frames).

Some examples comparing SAM2's over-segmentation with the correct segmentation by SAM2-LongVideo are shown in Figure 6.

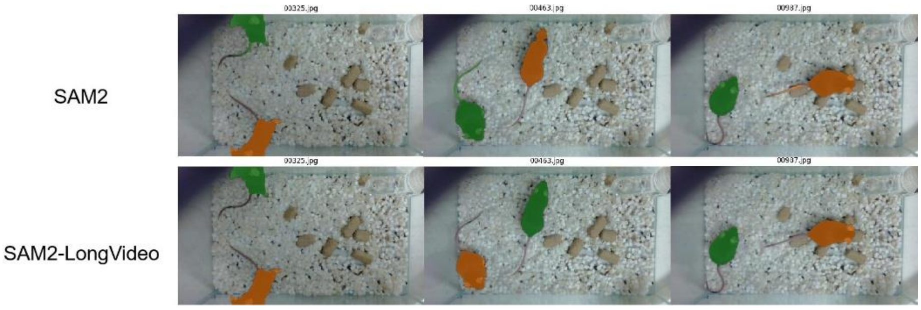

Fig. 6. Some examples comparing SAM2's over-segmentation with the correct segmentation by SAM2-LongVideo

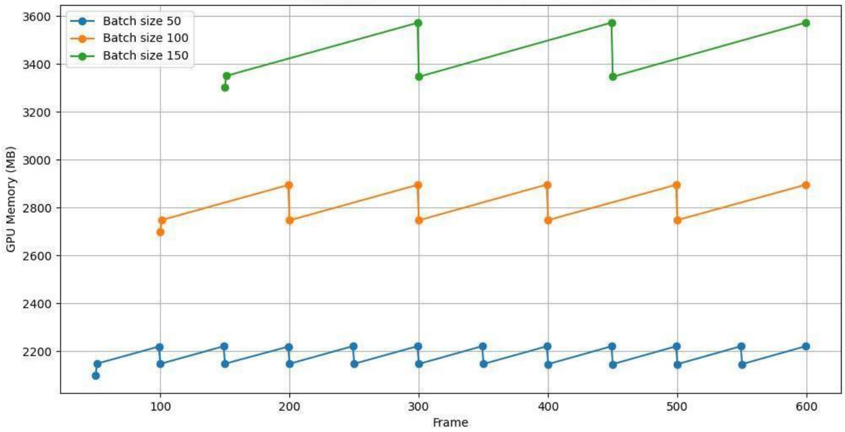

Fig. 7. GPU Memory Usage across Frames for Different Batch Sizes

This paper also qualitatively measured the GPU memory usage of SAM2-LongVideo. The model was tested on different videos and it was found that GPU memory usage is only related to the image resolution, and not affected by the number of tracked objects. Tests were conducted with image resolutions ranging from 320×180 to 1980×1080, and the memory usage was found to fluctuate by only ±5MB.

Fig.7 shows the GPU memory usage under different conditions, using a 1280x720 image sequence. The batch size in the figure refers to the size of the sub-sequences, excluding the first sub-sequence. The first sub-sequence is usually used for training the YOLO detection head, so it contains more images than the subsequent sub-sequences.

The results represent the GPU memory usage after the YOLO detection head has been trained. Each sub-sequence is provided with a prompt and segmented sequentially. As discussed earlier, the typical batch size range is 50-150. With 1280x720 images, the GPU memory usage ranges from 2.2GB to 3.5GB, which is very convenient for most users. Therefore, this method is suitable for computationally constrained scenarios without compromising segmentation performance.

Besides, using SAM2 to process 1,000 frames requires 17 GB of GPU memory and takes 459.11 seconds. However, with SAM2-LongVideo and a batch size of 50, it only requires less than 3 GB and takes 455.11 seconds.

4 Conclusion

A core challenge in VOS is how to efficiently process a large number of frames. Segmentation speed and accuracy are undoubtedly the two most important factors influencing the widespread adoption of a VOS model. For high-precision supervised learning, it is typically necessary to obtain a labeled training set for specific scenarios. Therefore, current researchers are now focusing their attention on large visual models. SAM2 is such a good model which has a general segmentation capability. YOLO is renowned for its real-time detection capability, which is why it has been widely applied. As a result, these two models are integrated to form SAM2-LongVideo thus long videos can be segmented continuously in resource-constrained situations.

In this method, long video sequences are divided into many short sub-sequences, each of which is prompted by the YOLO detection head. YOLO detects the target object in each image of every sub-sequence sequentially. When the confidence score of a detection result in one image exceeds a pre-set threshold, the test result of this image is used as the prompt.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. N. Ravi, V. Gabeur, Y.-T. Hu, et al. SAM 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714, 2024.
2. H. Liu, E. Zhang, J. Wu, et al. Surgical SAM 2: Real-time segment anything in surgical video by efficient frame pruning. arXiv preprint arXiv:2408.07931, 2024.
3. J. Redmon, S. Divvala, R. Girshick, et al. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2016.
4. G. Jocher, J. Qiu. Ultralytics YOLOv11 (Version 11.0.0). Online. Available: https://github.com/ultralytics/ultralytics, 2024.
5. Freestyle&HTM World Championship. Dancing dogs are so cute, World Championship 2016 Moscow. Online video. Available: https://youtu.be/GL3DXJE9UJk, 2016.
6. Y. Zhou, G. Sun, Y. Li, et al. When SAM2 meets video camouflaged object segmentation: A comprehensive evaluation and adaptation. arXiv preprint arXiv:2409.18653, 2024.