SVAC: Scaling Is All You Need For Referring Video Object Segmentation

Li Zhang[1]
lz2714@columbia.edu
oxiang Gao[2]
xiang@alumni.cmu.edu
hao Zhang[1]
ihao@columbia.edu
xiao Huang[3]
583@nyu.edu
Zhang[4*]
ng_tao@whu.edu.cn

[1] Columbia University
New York, USA
[2] Carnegie Mellon University
Pittsburgh, USA
[3] New York University
New York, USA
[4] Wuhan University
Wuhan, China

Abstract

Referring Video Object Segmentation (RVOS) aims to segment target objects in video sequences based on natural language descriptions. While recent advances in Multi-modal Large Language Models (MLLMs) have improved RVOS performance through enhanced text-video understanding, several challenges remain, including insufficient exploitation of MLLMs' prior knowledge, prohibitive computational and memory costs for long-duration videos, and inadequate handling of complex temporal dynamics. In this work, we propose **SVAC**, a unified model that improves RVOS by scaling up input frames and segmentation tokens to enhance video-language interaction and segmentation precision. To address the resulting computational challenges, **SVAC** incorporates the **Anchor-Based Spatio-Temporal Compression (ASTC)** module to compress visual tokens while preserving essential spatio-temporal structure. Moreover, the **Clip-Specific Allocation (CSA)** strategy is introduced to better handle dynamic object behaviors across video clips. Experimental results demonstrate that **SVAC** achieves state-of-the-art performance on multiple RVOS benchmarks with competitive efficiency. Our code is available at https://github.com/lizhang1998/SVAC.

Introduction

Referring Video Object Segmentation (RVOS) [10, 14, 17, 32] is a rapidly evolving task that aims to segment target objects in a video sequence based on natural language descriptions. Traditional RVOS methods [10, 17, 25, 48, 51, 54, 56] focus on designing effective fusion strategies for text and video features to align textual queries with visual content. Recent advancements [3, 18, 44, 49] leveraging Multi-modal Large Language Models (MLLMs) have revolutionized RVOS, achieving superior performance through enhanced understanding and

* Indicates corresponding author.

reasoning capabilities over text and video interactions. By integrating multi-modal inputs into a unified framework, MLLMs enable robust alignment of dynamic visual content with complex textual descriptions, setting new benchmarks in RVOS.

Despite the remarkable progress driven by MLLM-based models like Sa2VA [49] and its predecessors, several critical challenges persist, limiting the scalability and robustness of RVOS. These include: **(1) Insufficient exploitation of MLLMs' prior knowledge**: Existing approaches [18, 44, 49] often under-utilize the rich prior knowledge within MLLM due to feeding it only a sparse selection of frames. The resulting limited spatial and temporal information hinders the MLLM's ability to accurately locate objects referenced in the text descriptions, as shown in Fig. 1 (a); **(2) Prohibitive computational cost and memory usage for high-resolution, long-duration videos**: Processing high-resolution video inputs places a significant burden on MLLM architectures, especially Transformer-based models with their quadratic attention complexity. Furthermore, maintaining temporal coherence across extended video sequences necessitates the costly storage and updating of extensive feature maps or hidden states, resulting in substantial memory overhead; **(3) Inadequate handling of complex temporal dynamics**: Current methods [3, 18, 44, 49] rely on a single segmentation token to represent object localization throughout the entire video. However, for objects in rapid motion, a single token is insufficient to capture dynamic changes, as it cannot adapt to varying object positions and appearances across frames (see in Fig. 1 (b)).

In this work, regarding *challenge (1)*, we observe that scaling up video frames significantly enhances MLLM performance in RVOS. The previous SOTA model, Sa2VA, only samples 5 frames from the whole video, often failing to segment objects in the middle portion of the video. By scaling up to 10 or 20 frames by providing richer spatial and temporal cues, this issue is alleviated. However, as noted in *challenge (2)*, further increasing the frame number substantially increases computational and memory demands.

To overcome *challenge (2)*, the key lies in reducing the number of visual tokens before input to LLM. We propose a novel compression method called **Anchor-Based Spatio-Temporal Compression** (**ASTC** in Section 3.3) that optimizes the trade-off between processing a larger number of frames and preserving fine-grained details critical for accurate segmentation. Unlike existing pooling methods [23, 40], which often compromise on spatial resolution or temporal coherence, and in contrast to merging strategies [21, 46] that necessitate structural modifications to the MLLM, ASTC method divides videos into clips, retaining the first frame and compressing remaining frames into a single resized composite, preserving critical spatial-temporal features. It achieves superior memory efficiency, reduced computational complexity, and enhanced segmentation accuracy without requiring any change to the model structure, ensuring seamless plug-in compatibility.

Finally, to address the *challenge (3)* of complex object motion in videos, we scale up the number of segmentation tokens with the **Clip-Specific Allocation** (**CSA** in Section 3.4) strategy. Previous approaches [3, 18, 44, 49] typically use a single segmentation token per object through the entire video, which struggles with nuanced temporal dynamics such as rapid trajectory shifts or occlusions. Our method addresses this by assigning a dedicated segmentation token to each video clip. This refined strategy significantly enhances segmentation accuracy by more effectively capturing intricate motion patterns within these shorter temporal segments.

Based on the above discussions, we present **SVAC**: **S**caling up both **V**ideo frames with the **A**STC method and segmentation tokens with the **C**SA strategy. Built upon the Sa2VA framework, SVAC achieves SOTA performance on multiple RVOS benchmarks. Overall, our main contributions are summarized as follows:

- We discover that scaling up video frames input to MLLM significantly boosts its performance in RVOS.
- To mitigate the increased computational and memory demands, we propose Anchor-Based Spatio-Temporal Compression (ASTC, see Section 3.3) method to reduce visual tokens prior to being input to LLM. Additionally, to handle complex object motion in videos, we scale up segmentation tokens with Clip-Specific Allocation (CSA in Section 3.4) strategy, rather than leveraging a single token for the entire video.
- We develop a unified model named SVAC, and extensive experiments show that it achieves SOTA performance on multiple RVOS benchmarks.

2 Related Work

Multi-modal Large Language Models. The development of Multimodal Large Language Models (MLLMs) has accelerated rapidly in recent years, enabling unified understanding and reasoning across vision and language modalities. Early efforts [16, 29, 47, 50] such as CLIP [29] establish the foundation by learning joint image-text representations through contrastive learning. This is followed by models like Flamingo [1], which fuses vision inputs into frozen large language models (LLMs) using lightweight cross-attention modules. A key trend in MLLM design is the decoupling of vision encoders and language backbones. Vision modules such as ViT [12], EVA [13], or CLIP encoders are often paired with powerful LLMs, including LLaMA [35], ChatGLM [15], Qwen [45], Vicuña [9] and InternLM [6]. These LLMs provide strong language understanding and generation capabilities, and are typically aligned with vision features through learned projection layers or Q-formers, as demonstrated in BLIP-2 [20], LLaVA [26] and others [2, 7, 8, 19, 22, 36, 37, 38, 39, 42, 57]. Among these MLLMs, we choose InternVL2-5 [8, 11] as our backbone due to its superior performance in multimodal tasks such as VQA, image captioning, and document understanding.

Referring Video Object Segmentation. Referring Video Object Segmentation (RVOS) seeks to segment a target object throughout a video based on a natural language expression. Early approaches [10, 17, 25, 28, 48, 51, 54, 56] explore various fusion modules to integrate visual and linguistic features, achieving improved performance in generating accurate masks. Subsequent works [43] leverage transformer-based architectures to unify segmentation and tracking in video, enabling more robust handling of temporal dynamics. With the advent of MLLM, many works [18, 30, 49, 52, 53] attempt to introduce MLLM into the field of referring segmentation. For instance, LISA [18] proposes reasoning-based segmentation, while GLaMM [30] curates a new dataset to support region-level captioning and segmentation tasks. Similarly, VISA [44] enhances segmentation by selecting textrelevant frames for MLLM processing. More recently, Sa2VA [49] integrates SAM-2 [31] with MLLM, achieving strong performance across both image and video referring segmentation tasks, as well as visual question answering (VQA). However, existing methods are limited by processing only a small number of frames and relying on a single segmentation token, which constrains the capacity of MLLMs and hinders the modeling of fine-grained temporal dynamics. In contrast, our proposed SVAC model employs ASTC to efficiently process substantially more frames, fully harnessing the potential of MLLMs. Additionally, by scaling segmentation tokens through CSA, our method captures subtle temporal variations with higher precision.

Compression. Compression techniques for Vision Transformers (ViTs) and Video-Language

Models (VLMs) aim to reduce token redundancy while preserving performance. A notable approach, Token Merging (ToMe) [4], employs Bipartite Soft Matching to merge similar tokens across ViT layers. However, it often loses fine-grained details essential for tasks like Referring Video Object Segmentation (RVOS) and requires intrusive model modifications, reducing its flexibility. Subsequent works like VisionZip [46] and TokenPacker [21] explore alternative similarity metrics but face similar limitations in detail-sensitive applications. In contrast, methods like LongVLM [40] and LLaMA-VID [23] use average pooling on vision tokens post-encoding to achieve compression. While more adaptable, these approaches still struggle to retain critical details, making them less effective for RVOS. To address these challenges, we propose ASTC method, which ensures detail preservation in RVOS while maintaining computational efficiency.

3 Method

3.1 Task Description

In this section, we would describe RVOS task more specifically. Formally, given a video sequence $V = \{I_1, I_2, \ldots, I_T\}$, where each frame $I_t \in \mathbb{R}^{H \times W \times 3}$ represents an RGB image of spatial resolution $H \times W$ and temporal length T, and a referring expression S composed of L words, i.e., $S = \{w_1, w_2, \ldots, w_L\}$, the goal is to predict a sequence of binary segmentation masks $M = \{M_1, M_2, \ldots, M_T\}$, where each $M_t \in \{0,1\}^{H \times W}$ corresponds to the pixel-wise mask of the referred object in frame I_t.

The task can be formulated as learning a mapping function:

$$f : (V, S) \rightarrow M \tag{1}$$

where the model f learns to align the spatial-temporal visual features of the video V with the semantic representation of the referring expression S, to produce frame-level object masks that are consistent with both the visual context and the language description.

3.2 Overall Architecture

The overall structure is shown in Fig. 2.

Pretrained MLLM. The core of our structure is a pretrained MLLM, which consists of three key components: a vision encoder, a projection layer, and a large language model (LLM). The vision encoder processes input frames, compressed by ASTC model, to extract compact visual features. These features are then mapped by the projection layer into visual tokens that align with the text token embedding space, enabling seamless integration with text tokens. The combined sequence of visual and text tokens is fed into the LLM, which generates predictions by modeling the joint distribution of multimodal inputs.

Segmentation. To enable precise video segmentation, the system integrates SAM-2 with the MLLM using a list of <SEG> token as a bridge. The MLLM processes the input sequence and extracts the hidden states of the <SEG> tokens, which serve as a spatial-temporal prompt for the SAM-2 Decoder. This prompt provides contextual cues from the multimodal input to guide segmentation. Meanwhile, the SAM-2 Encoder extracts frame-level features from the input video frames and feeds them to the Decoder. By combining the MLLM's prompt with these frame features, the SAM-2 Decoder generates accurate, temporally consistent segmentation masks across the video.

(a) Please segment a man clapping and stepping back while next to the hula-hooping bear

(b) Please segment the horse whose head is being struck by the tail of another horse

Figure 1: **Visualization of Comparison between Sa2VA and our SVAC.** (a) Sa2VA shows poor contextual comprehension by segmenting a bear when explicitly instructed to segment a person, while our SVAC correctly localizes and segments the human subject as specified. (b)When confronted with complex motion patterns, such as two horses occluding each other and moving rapidly, Sa2VA's segmentation capabilities deteriorate significantly, whereas SVAC maintains accurate segmentation performance.

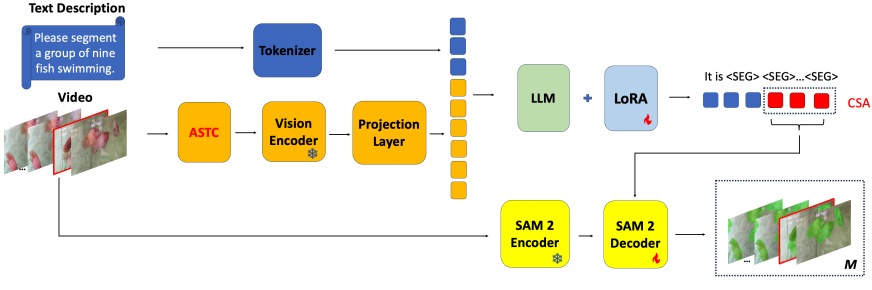

Figure 2: **Overall Structure.** The pipeline consists of three main stages: (1) Text and video frames are encoded and projected into a shared embedding space before being fed into the LLM; (2) <SEG> tokens are extracted from the LLM output; (3) The SAM-2 Decoder generates the final mask by combining features from the SAM-2 Encoder with the extracted <SEG> tokens.

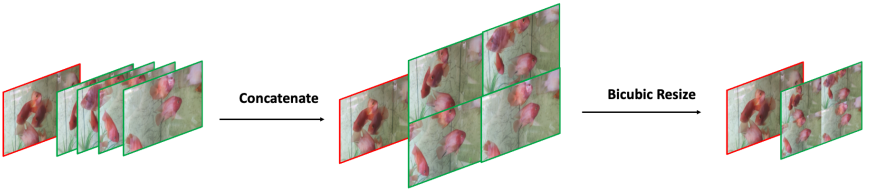

Figure 3: **Anchor-Based Spatio-Temporal Compression (ASTC).** For a 5-frame video clip, we preserve the first frame as anchor while compressing the remaining 4 frames through temporal concatenation (left-to-right, top-to-bottom arrangement) and subsequent resizing to match the anchor frame dimensions.

3.3 Scaling Frames

During testing with the Sa2VA model, we observe that increasing the number of sampled frames per video significantly improves performance. Initially, Sa2VA samples only five frames, but scaling to more frames causes out-of-memory issues. To mitigate the computational burden of processing dense visual tokens while preserving critical spatio-temporal information, we introduce **Anchor-Based Spatio-Temporal Compression** (**ASTC**), as shown in Fig. 3.

Given a video sequence $V = \{I_1, I_2, \ldots, I_T\}$, where each frame $I_t \in \mathbb{R}^{H\times W\times 3}$ represents an RGB image at time step t, we first partition the sequence into N non-overlapping clips of equal temporal length m. Each clip is denoted as $C^{(i)} = \{I_1^{(i)}, I_2^{(i)}, \ldots, I_m^{(i)}\}$ for $i = 1, \ldots, N$. Within each clip $C^{(i)}$, the first frame $I_1^{(i)}$ is designated as the anchor frame and retained at its full spatial resolution to anchor the spatial context. The subsequent frames $\{I_2^{(i)}, I_3^{(i)}, \ldots, I_m^{(i)}\}$, which capture the temporal evolution, undergo a dual compression process in both spatial and temporal dimensions.

First, the subsequent frames are concatenated along the spatial dimensions while preserving the channel dimension. The concatenation is performed in temporal order, arranging frames from left to right and top to bottom to preserve temporal coherence. To maintain the aspect ratio of the original frames $H \times W$, these $m-1$ frames are arranged into a 2D grid layout, forming an aggregated tensor:

$$I_{agg}^{(i)} = Concat(I_2^{(i)}, I_3^{(i)}, \ldots, I_m^{(i)}) \in \mathbb{R}^{H'\times W'\times 3} \tag{2}$$

Here, as defined in Equation 2, H' and W' are chosen such that the grid layout accommodates all $m-1$ frames while preserving an aspect ratio close to $H \times W$. If the spatial dimensions $H' \times W'$ slightly exceed what is required to exactly fit the frames, the remaining unused regions in the grid are filled with zeros (zero-padding) to maintain uniformity.

Next, applying bicubic interpolation to resize $I_{\text{agg}}^{(i)}$ back to the original spatial dimensions of a single frame:

$$\tilde{I}_{agg}^{(i)} = Bicubic\_Resize(I_{agg}^{(i)}, H \times W \times 3) \in \mathbb{R}^{H\times W\times 3} \tag{3}$$

This compressed representation $\tilde{I}_{\text{agg}}^{(i)}$ effectively encapsulates the temporal dynamics of frames $\{I_2^{(i)}, \ldots, I_m^{(i)}\}$ while reducing the spatial and channel footprint.

The final representation of each clip $C^{(i)}$ is the pair: $\{I_1^{(i)}, \tilde{I}_{\text{agg}}^{(i)}\}$, where $I_1^{(i)}$ serves as the anchor frame with full spatial fidelity, and $\tilde{I}_{\text{agg}}^{(i)}$ provides a compressed embedding of the temporal dynamics across the remaining frames.

Assuming a vision encoder generates s tokens per frame, the token count for the original clip is $s_{original} = m \cdot s$. After applying ASTC, the token count is reduced to $s_{reduced} = 2 \cdot s$, yielding a compression ratio of $2/m$. This approach significantly enhances computational efficiency and memory utilization while maintaining both the spatial detail of the anchor frame and the temporal context of the clip.

3.4 Scaling Tokens

In order to enhance the model's ability to distinguish between different temporal segments, we introduce **Clip-Specific Allocation** (**CSA**) to scale up the number of <SEG> tokens: Given a video that has been divided into N clips $\{C^{(1)}, C^{(2)}, ..., C^{(N)}\}$, we assign a unique segment token <SEG>$^{(i)}$ to each clip $C^{(i)}$.

Formally, let $S = \{\texttt{<SEG>}^{(1)}, \texttt{<SEG>}^{(2)}, ..., \texttt{<SEG>}^{(N)}\}$ be the set of clip-specifc segmentation tokens, where each <SEG>$^{(i)} \in \mathbb{R}^d$ and d is the embedding dimension. Instead of using a single, shared <SEG> token for all frames as a global prompt, we concatenate all the segmentation tokens and input them to SAM-2. For the frame $I_t^{(i)}$, which denotes the frame at time step t within clip $C^{(i)}$, we compute the mask M_t as follows:

$$M_t = SAM2\_Decoder\big(SAM2\_Encoder(I_t^{(i)}),\ \texttt{<SEG>}^{(i)}\big) \tag{4}$$

By providing temporally-aware <SEG> tokens, the model can better localize objects and segment changes over time, as each <SEG> token encodes clip-specific temporal priors and scene context.

Table 1: Referring Video Object Segmentation Performance

Method	Year	MeVIS$_{val_u}$ [10]			Ref-DAVIS17 [17]			ReVOS [42]		
		J	*F*	*J&F*	*J*	*F*	*J&F*	*J*	*F*	*J&F*
URVOS [33]	ECCV 2020	-	-	-	47.3	56.0	51.6	-	-	-
MTTR [6]	CVPR 2022	-	-	-	-	-	-	25.1	25.9	25.5
ReferFormer [41]	CVPR 2022	-	-	-	58.1	64.1	61.1	26.2	29.9	28.1
LISA-13B [18]	CVPR 2024	-	-	-	63.2	68.8	66.0	39.8	43.5	41.6
TrackGPT-13B [53]	ArXiv2024	-	-	-	62.7	70.4	66.5	43.2	46.8	45.0
VISA-13B [42]	ECCV 2024	-	-	-	67.0	73.8	70.4	48.8	52.9	50.9
VideoLISA [8]	NeurIPS 2024	50.9	58.1	54.5	72.7	64.9	68.8	-	-	-
VideoGLaMM [27]	CVPR 2025	-	-	-	73.3	65.6	69.5	-	-	-
GLUS [22]	CVPR 2025	-	-	61.6	-	-	-	47.5	52.4	49.9
ViLLa [55]	ArXiV 2025	-	-	-	70.6	78.0	74.3	54.9	59.1	57.0
Sa2VA-1B [49]	ArXiV 2025	-	-	53.4	-	-	69.5	-	-	47.6
Sa2VA-4B [49]	ArXiV 2025	-	-	55.9	-	-	73.7	-	-	53.2
Sa2VA-8B [49]	ArXiV 2025	-	-	58.9	-	-	75.9	-	-	57.6
Sa2VA-26B [49]	ArXiV 2025	-	-	61.8	-	-	78.6	-	-	58.4
SVAC-1B (Ours)	BMVC 2025	55.8	64.6	60.2	67.5	76.1	71.8	48.1	50.7	49.4
SVAC-4B (Ours)	BMVC 2025	56.9	65.6	61.3	71.4	80.0	75.7	55.1	56.1	55.6
SVAC-8B (Ours)	BMVC 2025	**58.2**	**67.3**	**62.8**	**74.9**	**83.8**	**79.4**	**58.2**	**60.2**	**59.2**

4 Experiments

4.1 Experimental Settings

BaseLine. We construct the baseline by integrating the SOTA MLLM models like InternVL2-5 [8] and SAM-2 [31], following the Sa2VA [49] framework. We sample 100 frames per video and apply the ASTC method before feeding them to the MLLM. In ASTC, frames are divided into 10 clips, and clip-specific segmentation allocation generates 10 <SEG> tokens, whose hidden states are processed by the SAM-2 decoder to produce the final mask.
Dataset & Metrics. We fine-tune on three major RVOS datasets: MeVIS [10] dataset, Ref-YoutubeVOS [32] dataset, and ReVOS [44] dataset. Performance is evaluated using the J(average IoU), F (boundary F measure), and $J\&F$(average of J and F) metric.
Implementation Details. Training and testing are conducted using the XTuner codebase. The initial learning rate is set to 4e-5, with the perception model frozen and the LLM fine-tuned using LoRA. The LLM's maximum sequence length is 12,288. Each video is sampled at 100 frames, grouped into clips of 10 frames each. All experiments are performed on 8 NVIDIA A800 GPUs with 40GB of memory.

4.2 Main Results

Table 1 demonstrates the SOTA performance of our SVAC model across multiple benchmarks. With only 8B parameters, SVAC achieves SOTA $J\&F$ scores of 62.8, 79.4, and 59.2 on MeVIS(val_u) [10], Ref-DAVIS17 [17], and ReVOS [44] datasets respectively, surpassing the previous SOTA model Sa2VA-26B by margins of 1.0, 0.8, and 0.8 points.

4.3 Ablation Study

To better understand the impact of each component in our framework, we conduct a series of ablation studies. Unless otherwise specified, all experiments are performed on the MeVIS dataset using the 1B variants of the MLLM backbone.

Compression Method. We compare our proposed ASTC method againt several common compression techniques, including pooling [23, 40], token pruning [34], token merging [4, 21, 46]. For all methods, we set a compression ratio of 25% and sample 80 frames. The evaluated methods are: (1) Average Pooling with 2×2 kernel and stride of 2; (2) Max Pooling with 2×2 kernel and stride of 2; (3) Token Pruning, retaining only the top 25% of tokens based on attention scores to the <CLS> token; (4) Token Merging, uniformly sampling 25% of tokens and merging the remaining tokens into them; (5) ASTC, configured with 8 frames per clip. As shown in Table 2, ASTC outperforms all other methods by at least 1 point.

Scaling Curve of Frames. We further evaluate how the number of frames per video affects performance. Using the ASTC method with clips of 10 frames each, we increase the sampled frames from 10 to 100 in increments of 10, maintaining a compression ratio of 20%. As illustrated in Fig. 4, with the increasing number of frames, J, F and $J\&F$ metrics consistently improve as the number of frames increases.

<SEG> Token Granularity. We analyze the effect of varying the number of clips per <SEG> token in a video with 100 sampled frames, compressed via ASTC into 10 clips of 10

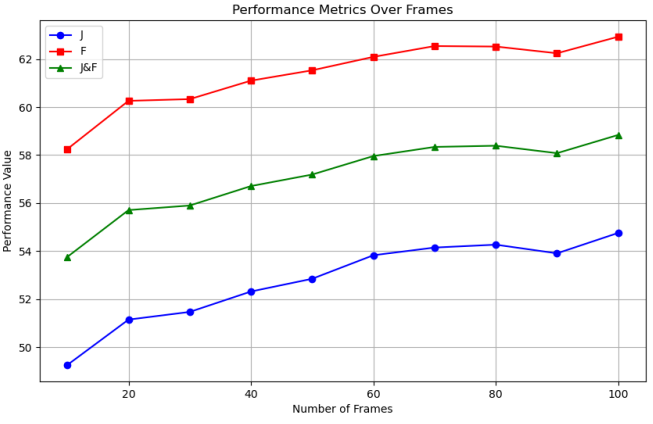

Figure 4: Scaling Curve of Frames

frames each. Specifically, we assign one <SEG> token per: (1) 10 clips (covering the entire video); (2) 5 clips (1 token per 20 frames); (3) 3 clips (approximately 1 token per 33 frames, with the last 4 clips sharing the same <SEG> token); (4) 2 clips (1 token per 50 frames); and (5) 1 clip (1 token per 10 frames). As shown in Table 3, assigning one <SEG> token per clip yields the best performance across the evaluated metrics - J, F and $J\&F$.

Method	J	F	$J\&F$
Average Pooling	52.9	60.7	56.8
Max Pooling	53.3	61.5	57.4
Token Prune	51.1	59.5	55.3
Token Merge	52.8	60.7	56.8
ASTC	54.4	62.6	58.5

Table 2: Compression Strategy Comparison

Clips per Token	J	F	$J\&F$
10	54.8	62.9	58.9
5	54.7	63.1	58.9
3	54.8	63.1	59.0
2	54.9	63.3	59.1
1	55.3	63.5	59.4

Table 3: Effect of <SEG> Token Granularity

4.4 Limitations and Future Work

Although our SVAC achieves SOTA performance, several limitations remain. First, our method may struggle with objects that appear briefly in the video sequence, as their temporal information might be lost during compression and attention computation. Second, challenging scenarios involving extreme occlusions or dramatic appearance changes can still pose difficulties, as the current number of segmentation tokens may not be sufficient to maintain consistent object tracking under such complex conditions.

Future work could address these limitations through several directions: (1) incorporating advanced motion prediction mechanisms to better handle brief object appearances and disappearances; (2) developing adaptive token allocation strategies that can dynamically adjust based on scene complexity; and (3) exploring multi-scale temporal feature fusion techniques to enhance the model's robustness to appearance variations.

5 Conclusion

In this work, we propose SVAC, a novel framework for Referring Video Object Segmentation (RVOS) that effectively scales up both the number of processed video frames and segmentation tokens. We introduce Anchor-Based Spatio-Temporal Compression (ASTC), which processes more frames to better leverage MLLM's potential by maintaining an anchor frame and a composite compressed representation of other frames for each clip, and Clip-Specific Allocation (CSA), which increases the number of segmentation tokens through clip-wise allocation to better capture motion patterns. Extensive experiments across multiple VOS benchmarks demonstrate that SVAC achieves state-of-the-art performance while maintaining computational efficiency. Our work provides a promising direction for advancing text-guided video understanding and segmentation tasks.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966.

[3] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Lei Liu, Zheng Zhang, and Mike Zheng Shou. One token to seg them all: Language instructed reasoning segmentation in videos, 2024. URL https://arxiv.org/abs/2409.19603.

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster, 2023. URL https://arxiv.org/abs/2210.09461.

[5] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye,

Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. Internlm2 technical report, 2024.

[7] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[8] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025. URL https://arxiv.org/abs/2412.05271.

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[10] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. MeViS: A large-scale benchmark for video segmentation with motion expressions. In *ICCV*, 2023.

[11] Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*, 2024.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[13] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19358–19369, 2023.

[14] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees G. M. Snoek. Actor and action video segmentation from a sentence. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5958–5966, 2018. URL https://api.semanticscholar.org/CorpusID:4002773.

[15] Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024. URL https://arxiv.org/abs/2406.12793.

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[17] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV 14*, pages 123–141. Springer, 2019.

[18] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9579–9589, June 2024.

[19] Weixian Lei, Jiacong Wang, Haochen Wang, Xiangtai Li, Jun Hao Liew, Jiashi Feng, and Zilong Huang. The scalability of simplicity: Empirical analysis of vision-language learning with a single transformer. *arXiv preprint arXiv:2504.10462*, 2025.

[20] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[21] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm, 2024. URL https://arxiv.org/abs/2407.02392.

[22] Xiangtai Li, Tao Zhang, Yanwei Li, Haobo Yuan, Shihao Chen, Yikang Zhou, Jiahao Meng, Yueyi Sun, Shilin Xu, Lu Qi, et al. Denseworld-1m: Towards detailed dense grounded caption in the real world. *arXiv preprint arXiv:2506.24102*, 2025.

[23] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. 2024.

[24] Lang Lin, Xueyang Yu, Ziqi Pang, and Yu-Xiong Wang. Glus: Global-local reasoning unified into a single large language model for video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[25] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *CVPR*, 2023.

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[27] Shehan Munasinghe, Hanan Gani, Wenqi Zhu, Jiale Cao, Eric Xing, Fahad Khan, and Salman Khan. Videoglamm: A large multimodal model for pixel-level visual grounding in videos. *ArXiv*, 2024. URL https://arxiv.org/abs/2411.04923.

[28] Quanzhu Niu, Yikang Zhou, Shihao Chen, Tao Zhang, and Shunping Ji. Beyond appearance: Geometric cues for robust video instance segmentation. *arXiv preprint arXiv:2507.05948*, 2025.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[30] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M. Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S. Khan. Glamm: Pixel grounding large multimodal model. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[31] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[32] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 208–223. Springer, 2020.

[33] Nicholas Stroh. Trackgpt – a generative pre-trained transformer for cross-domain entity trajectory forecasting, 2024. URL https://arxiv.org/abs/2402.00066.

[34] Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. Tokencarve: Information-preserving visual token compression in multimodal large language models, 2025. URL https://arxiv.org/abs/2503.10501.

[35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

[36] Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, et al. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology. *arXiv preprint arXiv:2507.07999*, 2025.

[37] Jiacong Wang, Bohong Wu, Haiyong Jiang, Xun Zhou, Xin Xiao, Haoyuan Guo, and Jun Xiao. World to code: Multi-modal data generation via self-instructed compositional captioning and filtering. *arXiv preprint arXiv:2409.20424*, 2024.

[38] Jiacong Wang, Zijiang Kang, Haochen Wang, Haiyong Jiang, Jiawen Li, Bohong Wu, Ya Wang, Jiao Ran, Xiao Liang, Chao Feng, et al. Vgr: Visual grounded reasoning. *arXiv preprint arXiv:2506.11991*, 2025.

[39] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191.

[40] Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models, 2024. URL https://arxiv.org/abs/2404.03384.

[41] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. *arXiv preprint arXiv:2201.00487*, 2022.

[42] Xin Xiao, Bohong Wu, Jiacong Wang, Chunyuan Li, Haoyuan Guo, et al. Seeing the image: Prioritizing visual correlation by contrastive alignment. *Advances in Neural Information Processing Systems*, 37:30925–30950, 2024.

[43] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *CVPR*, 2023.

[44] Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. Visa: Reasoning video object segmentation via large language models. *arXiv preprint arXiv:2407.11325*, 2024.

[45] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

[46] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*, 2024.

[47] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cpDhcsEDC2.

[48] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018.

[49] Haobo Yuan, Xiangtai Li, Tao Zhang, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. *arXiv*, 2025.

[50] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision, 2021. URL https://arxiv.org/abs/2111.11432.

[51] Tao Zhang, Xingye Tian, Yu Wu, Shunping Ji, Xuebo Wang, Yuan Zhang, and Pengfei Wan. Dvis: Decoupled video instance segmentation framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1282–1291, 2023.

[52] Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding. *Advances in Neural Information Processing Systems*, 37:71737–71767, 2024.

[53] Tao Zhang, Xiangtai Li, Zilong Huang, Yanwei Li, Weixian Lei, Xueqing Deng, Shihao Chen, Shunping Ji, and Jiashi Feng. Pixel-sail: Single transformer for pixel-grounded understanding. *arXiv preprint arXiv:2504.10465*, 2025.

[54] Tao Zhang, Xingye Tian, Yikang Zhou, Shunping Ji, Xuebo Wang, Xin Tao, Yuan Zhang, Pengfei Wan, Zhongyuan Wang, and Yu Wu. Dvis++: Improved decoupled framework for universal video segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[55] Rongkun Zheng, Lu Qi, Xi Chen, Yi Wang, Kun Wang, Yu Qiao, and Hengshuang Zhao. Villa: Video reasoning segmentation with large language model, 2025. URL https://arxiv.org/abs/2407.14500.

[56] Yikang Zhou, Tao Zhang, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Improving video segmentation via dynamic anchor queries. In *European Conference on Computer Vision*, pages 446–463. Springer, 2024.

[57] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.