One Token to Seg Them All: Language Instructed Reasoning Segmentation in Videos

Zechen Bai[1]   **Tong He**[2]   **Haiyang Mei**[1]   **Pichao Wang**[2]   **Ziteng Gao**[1]
Joya Chen[1]   **Lei Liu**[2]   **Zheng Zhang**[2]   **Mike Zheng Shou**[1]*
[1]Show Lab, National University of Singapore   [2]Amazon

Abstract

We introduce VideoLISA, a video-based multimodal large language model designed to tackle the problem of language-instructed reasoning segmentation in videos. Leveraging the reasoning capabilities and world knowledge of large language models, and augmented by the Segment Anything Model, VideoLISA generates temporally consistent segmentation masks in videos based on language instructions. Existing image-based methods, such as LISA, struggle with video tasks due to the additional temporal dimension, which requires temporal dynamic understanding and consistent segmentation across frames. VideoLISA addresses these challenges by integrating a Sparse Dense Sampling strategy into the video-LLM, which balances temporal context and spatial detail within computational constraints. Additionally, we propose a One-Token-Seg-All approach using a specially designed <TRK> token, enabling the model to segment and track objects across multiple frames. Extensive evaluations on diverse benchmarks, including our newly introduced ReasonVOS benchmark, demonstrate VideoLISA's superior performance in video object segmentation tasks involving complex reasoning, temporal understanding, and object tracking. While optimized for videos, VideoLISA also shows promising generalization to image segmentation, revealing its potential as a unified foundation model for language-instructed object segmentation. Code and model will be available at: https://github.com/showlab/VideoLISA.

1   Introduction

We live in a dynamic world. Localizing objects of interest in videos according to human intent is a crucial task for intelligent models and systems. Language, as a natural interface, serves as the primary reference for identifying target objects. However, language expressions vary widely across different scenarios, presenting varying levels of difficulty. While category names are straightforward references, detailed text descriptions from tasks like referring segmentation [28, 48, 59] introduce greater complexity. In real-world applications, these expressions can be more complex, involving intent understanding, reasoning, and world knowledge, making them more user-friendly yet significantly more challenging for models to understand and act upon.

Recent advancements in the image domain have shown progress in language-instructed reasoning for detection and segmentation tasks. Models leveraging multimodal large language models (MLLMs), such as those in DetGPT [53] and LISA [31], have demonstrated the ability to localize target objects by harnessing the implicit reasoning capabilities and world knowledge embedded in large language models (LLMs). However, these advancements have not seamlessly translated to video tasks, particularly video object segmentation (VOS). The primary challenge in VOS stems from the additional temporal dimension, which introduces complexities absent in static images. VOS requires

---

*Corresponding Author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

models to 1) on the input side, capture and comprehend the temporal dynamics present in the video; and 2) on the output side, predict temporally consistent segmentation masks across frames. These challenges render existing image-based methods inadequate for handling video tasks.

In this work, we introduce VideoLISA, a video-based MLLM designed to address language-instructed reasoning segmentation in videos. Our goal is to segment target objects throughout the entire video based on diverse language queries that necessitate scene understanding, temporal comprehension, and implicit reasoning. Drawing inspiration from previous works [53, 31], we employ an LLM to inherit its complex reasoning capabilities and adopt the Segment Anything Model (SAM) [30] to produce segmentation masks. To overcome the unique challenges presented by video data, we propose two key innovations: a Sparse Dense Sampling strategy and a One-Token-Seg-All approach.

To equip the model with video temporal understanding ability, it is necessary to involve multiple frames. Processing visual features from all sampled frames in full feature resolution is computationally prohibitive due to the large number of tokens. In pursuit of efficiency, reducing the frame number would limit the perception of temporal dynamics while down-sampling frame features would lose visual details that are essential for dense prediction tasks exemplified by segmentation. Our intuition is that adjacent frames in videos usually share similar visual contents and features. Therefore, we leverage this inherent *temporal redundancy* in videos and propose the Sparse Dense Sampling strategy. It uniformly samples a set of dense frames, preserving full-resolution features (*dense* tokens), and down-samples the remaining interleaved frames to lower resolution (*sparse* tokens). Dense tokens provide detailed visual information needed for accurate segmentation, while sparse tokens capture the temporal context, ensuring that the model remains aware of motion and changes over time. This balance allows the model to construct a coherent spatiotemporal narrative without excessive computational demands.

For achieving temporal consistency in segmentation, instead of handling separate representations for each frame, we propose a One-Token-Seg-All approach. Prior arts [21, 7] reveal that one compact representation can potentially associate the same object across video frames. In this work, we design a special <TRK> token to segment and track target objects across multiple frames. Specifically, we incorporate the <TRK> token into the model's vocabulary and utilize its last hidden embedding in the LLM to prompt the mask decoder to produce segmentation masks. We improve the temporal consistency from two aspects. First, when generating the <TRK> token, the model 'sees' the video content through the temporal module, which serves as the information foundation for cross-frame association. In addition, during training, the <TRK> token is intentionally trained to segment multiple frames simultaneously, preventing the model from learning shortcuts that focus only on spatial information of a certain frame. During inference, a single <TRK> token can segment and track objects across an entire video. The <TRK> token acts as a unified spatiotemporal representation, encapsulating object information across multiple frames and reducing the complexity of handling multiple prompts.

We evaluate our model on a comprehensive range of public benchmarks, including standard video/image referring segmentation, motion-guided video segmentation, and image reasoning segmentation. To further assess the model's capabilities in complex reasoning, temporal understanding, and object tracking, we introduce the ReasonVOS benchmark. Extensive experiments and ablation studies demonstrate the effectiveness of our approach. Although our model is particularly designed for videos, experiments show that it generalizes well on images, making it a potential foundation model for unified language instructed object segmentation. Our contributions are:

- Sparse Dense Sampling Strategy: We devise a sampling strategy for video-LLM training that achieves a balance between temporal context length and spatial visual detail under computational constraints. This strategy is shown to be effective for spatiotemporal dense prediction tasks, exemplified by video object segmentation.

- One-Token-Seg-All Approach: We design an effective approach for temporal consistent object segmentation in videos by utilizing a special <TRK> token. This strategy demonstrates robust performance in video object segmentation, leveraging the video-LLM learning module and a specially designed training objective.

- VideoLISA Model: We propose VideoLISA, a video-LLM that democratizes reasoning segmentation to videos. Additionally, we introduce the ReasonVOS benchmark, focusing on complex reasoning, temporal understanding, and object movements. This benchmark, along with a range of public benchmarks, comprehensively validates our model's performance.

2

2 Related Work

2.1 Video Object Segmentation

In computer vision, video object segmentation is a well-studied task [68]. Specifically, referring video object segmentation (RVOS) aims to segment the target object mentioned in a natural language expression in a video [59, 8, 63, 69, 35, 64, 49]. Compared with image segmentation, RVOS is more challenging since both the action and appearance of the referred object must be segmented in a video. Gavrilyuk et al. (2018) were the first to propose the RVOS task and the A2D-Sentences benchmark [23]. This field continues to evolve with new benchmarks emerge such as Ref-DAVIS-17 [29], Ref-YouTube-VOS [59], and MeViS [14]. Many previous studies have primarily adapted referring image segmentation approaches for frame-by-frame object segmentation. For example, URVOS [59] and RefVOS [6] utilize cross-modal attention for per-frame segmentation. Some recent works, such as ReferFormer [64] and MTTR [8], employ a DETR-like structure, which simplifies the referring pipeline and achieves impressive performance. R2VOS [35] enhances multi-modal alignment through text reconstruction. OnlineRefer [63] proposes an online model with explicit query propagation. SgMg [49] proposes a segment-and-optimize paradigm to solve the feature drift issue. Despite the impressive results achieved by these methods, several challenges remain. First, most existing methods are deficient in comprehending the motion information in videos and languages, as revealed by the recent MeViS [14] benchmark. Second, there are few studies on complex reasoning-based segmentation in the video domain, both methodologically and benchmark-wise.

2.2 Multimodal Large Language Model

The remarkable advancements of large language models (LLMs) motivate the research community to extend the foundational capabilities of LLMs to the visual domain, leading to multimodal large language models (MLLMs) [70, 4]. The pioneering works of MLLMs, such as LLaVA [41], MiniGPT-4 [74], and InstructBLIP [13], exhibit impressive visual understanding capabilities, including image captioning [60, 3] and visual question answering. When extending into the video domain, a prominent issue is handling the temporal dimension. One straightforward approach is to concatenate the tokens from multiple frames [39], though the temporal length might be limited by computational resources. To address this, one line of work [47, 27, 26] explores pooling (merging) strategies to reduce the number of tokens, such as pooling along the spatial and temporal dimensions separately [47], token merging based on similarity [27], and pooling with different strengths at a slow-fast pace [26]. Another line of work [34, 71, 36] utilizes the Q-former [33] architecture to extract abstracted features, which greatly reduces the number of tokens.

More recently, some studies have further integrated region-level image understanding and grounding abilities into MLLMs. Kosmos-2 [52] and Shikra [10] directly quantize bounding boxes into discrete location tokens or numeric representations of positions. GPT4RoI [72] uses a simple pooling operation to extract features within boxes or masks as the region representations. Another line of work leverages the reasoning ability of MLLMs and resorts to off-the-shelf models for localization. For example, DetGPT [53] utilizes a pre-trained LLM and an open-vocabulary object detector to detect the target object based on human intent described in natural language. LISA [31] proposes the task of image reasoning segmentation and connects an MLLM and the Segment Anything (SAM) [30] model using a special token to produce fine-grained segmentation masks. Built based on LISA, PixelLM [58] exhibits unique advantages in handling multiple objects in the reasoning segmentation task. Although these works have achieved impressive performance in image tasks, they are still incapable of processing videos. For object segmentation in videos, very few studies have leveraged the reasoning ability of LLMs to overcome current limitations. PG-Video-LLaVA [51] utilizes off-the-shelf object detector and tracker to obtain the target objects first and then match it with the entities mentioned in the generated text. TrackGPT [75] makes a straightforward extension of LISA by iteratively updating the special token with video progresses. However, the absence of video learning module significantly limits its perception and reasoning of temporal dynamics.

3 Method

The task of language-instructed reasoning segmentation in videos can be formally defined as follows. Given a video $\mathcal{X}_{\text{vid}}$ and a language expression $\mathcal{X}_{\text{txt}}$, the model takes both as input and outputs the

3

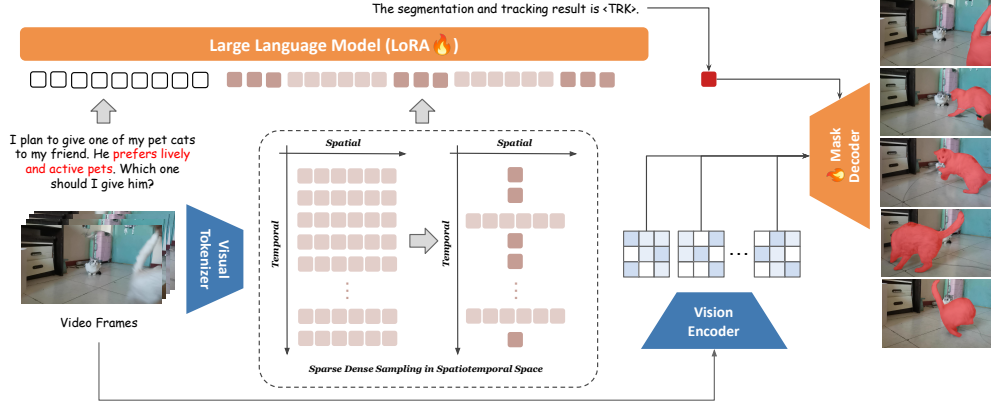



Figure 1: Framework of our approach.

pixel-level segmentation masks $\mathcal{M}$ for all frames. $\mathcal{X}_{\text{txt}}$ is a free-form text that particularly emphasizes implicit intent reasoning, world knowledge, and video temporal dynamics.

3.1 Architecture

Fig. 1 illustrates the model architecture. It consists of a visual tokenizer, an LLM, a vision encoder, and a promptable mask decoder. We omit the text tokenizer in the LLM for simplicity. The visual tokenizer and LLM are initialized from LLaVA [41, 55]. The vision encoder and mask decoder are initialized from SAM [30]. Given a video, we first uniformly sample T_{sparse} frames and encode them into visual tokens via the visual tokenizer, resulting in $T_{\text{sparse}} \times L$ tokens in total. Ideally, larger T_{sparse} would be better for capturing temporal dynamics. However, it is prohibitive to let the LLM process such a large number of tokens. Thus, we develop the Sparse Dense Sampling strategy to reduce the number of tokens, which will be elaborated in Sec. 3.2. After that, the visual tokens are concatenated with text tokens and fed into the LLM.

To equip the LLM with segmentation capabilities, following previous work [31], we extend the vocabulary of the LLM with a special token <TRK>. During generation, this special token carries rich semantic information from the text prompt and video content, providing signals for decoding pixel-level segmentation masks. Specifically, we extract the last layer embedding corresponding to the <TRK> token and transform it into a prompt embedding with a multi-layer perceptron (MLP). At the same time, the vision encoder extracts per-frame features from the video. Finally, the prompt embedding and the visual features are processed by the mask decoder to produce the segmentation masks. Note that for one video, there is only one prompt embedding that is in charge of all the frames. The One-Token-Seg-All approach will be introduced in Sec. 3.3.

3.2 Sparse Dense Sampling

Given the $T_{\text{sparse}} \times L$ tokens, we aim to reduce the number of tokens while preserving enough spatial details and temporal dynamics. Therefore, we further sample T_{dense} frames out of T_{sparse} frames. The visual tokens of the T_{dense} frames are all preserved in full resolution, *i.e., dense* tokens. Then, we apply global average pooling on the T_{sparse} frames to reduce them to low resolution, *i.e., sparse* tokens. In our implementation, each frame is represented by only one token. Finally, the total number of tokens is reduced to $T_{\text{sparse}} + T_{\text{dense}} \times L$, which is significantly smaller than $T_{\text{sparse}} \times L$. The rationale behind this strategy is the inherent temporal redundancy in video data. By exploiting this, we reduce the computational burden without losing critical information. The dense tokens provide visual details for their adjacent sparse frames, while the sparse tokens capture the temporal dynamics for the dense frames. In Sec. 2.2, we have discussed several popular temporal learning strategies in video-LLM. Although they exhibit remarkable performance in general video understanding tasks, our empirical studies (see Tab. 5) demonstrate that these popular strategies are not seamlessly transferable to video object segmentation. This is likely because they either lose spatial details or temporal information, both of which are essential in dense prediction tasks in videos.

4

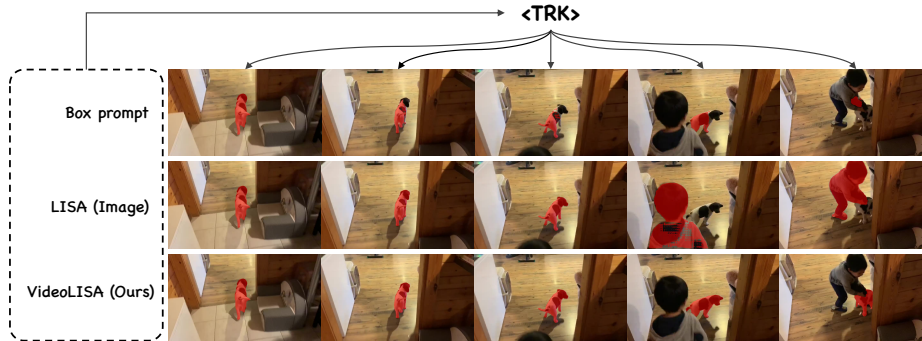



Figure 2: Exploration of One-Token-Seg-All approach.

3.3 One Token Seg All

As shown in Fig. 1, throughout the video, we use a single special <TRK> token for segmenting all the frames. We provide an in-depth analysis of the rationale behind this approach. In our model, the promptable segmentation model is initialized from SAM, in which the decoder takes the prompt embedding and visual features as inputs and outputs masks. Our intuition is that *segmenting one object in multiple frames can be regarded as segmenting multiple regions (instances) in one image grid*. From this perspective, SAM [30] itself already has the potential to segment objects across multiple frames, *if the prompt is properly given*. Previous works [21, 7, 20] have shown that one compact representation has the potential to associate the same entity across video frames. For example, from the perspective of object tracking [7], the prompt embedding can be regarded as a semantic kernel while the visual features are the context to be contrasted. This motivates us to explore whether one prompt embedding is capable of tracking under the promptable decoding paradigm of SAM.

To answer this question, one key problem is whether the prompt embedding contains enough semantic information to serve as the kernel. In SAM, its own prompt encoder mainly accepts visual prompts, such as points, boxes, and masks. In videos, the object moves dynamically. Our pilot study in Fig. 2 shows that visual prompts quickly fail in the presence of object motion. This is expected since these visual prompts heavily rely on the object's spatial location. We then explore the prompt embedding produced by an image reasoning segmentation model, LISA [31], which employs a LLM and is trained with segmentation data. It can be expected that its prompt embedding should contain more semantic information, at least significantly more than that of the visually instructed prompt. The second row of Fig. 2 validates this hypothesis by applying one prompt embedding to multiple frames. Compared to box prompts, the prompt embedding from LISA shows improved resilience to object movement, as demonstrated in the first three frames. However, when the object's motion becomes larger and a distractor object appears, the segmentation fails again, drifting to another object nearby.

We identify two primary factors that account for the failure. Firstly, the input of LISA model only has one frame, which contains very limited temporal information. Therefore, the generated prompt embedding lacks the information required for cross-frame association. Secondly, during the training of LISA, the prompt embedding is trained to segment only one frame. This potentially allows it to learn a shortcut that merely encompasses positional information, rather than learning the semantic information that generalizes across frames. In our work, the approach of using one token to segment multiple frames has been developed by addressing these issues accordingly. Firstly, the Sparse Dense Sampling-based temporal learning module provides spatiotemporal information of the video. The model 'sees' the video content, which is the foundation of mask association. Furthermore, during training, we intentionally train the <TRK> token to segment multiple frames. This objective would enforce the token to learn more 'semantic' information that can be used as the semantic kernel and segment the target object across frames. The last row of Fig. 2 presents the segmentation and tracking produced by the <TRK> token in our VideoLISA.

3.4 Training and Inference

Training Data. The training data for our model mainly consists of two parts: 1) image segmentation and 2) video segmentation. For the image part, we follow the setting of LISA [31]. For the video data, we employ video object segmentation (VOS) and referring video segmentation data (RVOS).

5

During pre-processing, we fill the original category name or referring expression in the dataset into a template. For example: "USER: `<VIDEO>` `Can you segment {description} in this scene?` ASSISTANT: `Sure, it is <TRK>.`", where {`description`} is the placeholder to fill. For VOS data that contain videos with multi-class labels, we randomly choose one class and merge all the masks belonging to this class into one binary mask.

Training Objective. The model is trained end-to-end using the text generation loss $\mathcal{L}_{\text{txt}}$ and segmentation loss $\mathcal{L}_{\text{seg}}$. The segmentation loss consists of per-pixel binary cross-entropy (BCE) loss and DICE loss. The final loss is computed as the weighted sum of the three losses. For video training, we compute the segmentation loss on the sampled T_{dense} frames in parallel and average them.

Inference. During inference, given a video, T_{sparse} and T_{dense} frames are sampled similarly to training, except that the T_{dense} frames are uniformly sampled from T_{sparse} rather than randomly. After obtaining the `<TRK>` token from the LLM, we feed all the frames of the video into the mask decoder one by one, using the same `<TRK>` token to segment each frame, yielding a list of masks.

Post optimization. Among these frames, the T_{dense} frames are seen in full resolution by the model, making their segmentation masks more reliable and accurate. For the remaining frames, although the One-Token-Seg-All strategy exhibits impressive cross-frame segmentation performance, our empirical observations indicate it inevitably suffers from low mask quality, likely limited by the inherent capability of the SAM model. Thus, we employ post-optimization as an optional step to further enhance mask quality. Specifically, we take XMem++ [5] as the post-optimization approach. Compared to XMem [12], which propagates one mask through the video, XMem++ distinguishes itself by taking multiple 'reliable' masks as reference and inferring the masks of the remaining frames. This paradigm is naturally suitable for our method since the T_{dense} frames span uniformly across the video, providing long-range yet diverse masks as references.

4 Benchmark

The versatile abilities of our model can be evaluated using public benchmarks that assess various aspects. RVOS benchmarks [29, 59] evaluate temporal-related abilities, involving referring expression comprehension, video temporal understanding, and temporal consistent segmentation. Complex reasoning abilities can be assessed by the image-based reasoning segmentation benchmark [31]. However, there is still a lack of a benchmark that comprehensively evaluates the reasoning segmentation abilities of videos. Towards this goal, we have organized the *ReasonVOS* benchmark. Specifically, we annotate language expressions based on the videos and mask annotations from existing datasets, including MOSE [15], MeViS [14], VIPSeg [50], and BURST [2]. The criteria for data collection and annotation processes are as follows. Each language expression should encompass at least one of the following aspects: 1) complex reasoning, 2) world knowledge, 3) temporal dynamics. For the video and mask selection, objects with explicit movement are highly prioritized to evaluate the temporal consistency of masks. As a result, we manually annotated 105 samples as initial seed data. Following previous practices [14, 75], we use a LLM to rephrase the language expressions for augmentation and perform another round of human checking. The resulting ReasonVOS benchmark comprises 458 video-instruction-mask data samples. This benchmark is specifically designed for zero-shot evaluation purposes, as the reasoning ability is embedded in the LLM and can be triggered by existing image-based reasoning segmentation data.

5 Experiments

5.1 Experimental Setting

Datasets Our model is trained on a variety of segmentation datasets. The image-based datasets include 1) semantic segmentation: ADE20K [73], COCO-Stuff [9], PACO-LVIS [54], and PASCAL-Part [11]; 2) referring segmentation: refCLEF, refCOCO, refCOCO+ [28], and refCOCOg [48]; 3) reason segmentation: 239 ReasonSeg samples from LISA [31]. The video-based datasets we use include: 1) semantic VOS: YouTube-VOS [66]; 2) referring VOS: Refer-YouTube-VOS [59] and MeViS [14]. The evaluation benchmarks will be elaborated in the corresponding experiment sections.

Implementation Details We implement our model with LLaVA-Phi-3-V [55], a multimodal LLM based on Phi-3 [1] with 3.8B parameters. We adopt the vision encoder and mask decoder from SAM [30]. We conduct joint training using both image and video datasets. For video data, we set $T_{\text{sparse}} = 32$ and $T_{\text{dense}} = 4$ according to our GPU memory. For image data, we duplicate the images as pseudo video data. We freeze the visual tokenizer and vision encoder, train the LLM with

6

Table 1: The quantitative evaluation results on Refer-Youtube-VOS and Refer-DAVIS-17. In the table, **bold** denotes the best scores; underline denotes the second place.

Method	Year	Refer-Youtube-VOS			Refer-DAVIS-17		
		$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Traditional methods without reasoning ability							
URVOS [59]	2020	47.2	45.2	49.1	51.6	47.2	55.9
CMPC-V [42]	2021	47.5	45.6	49.3	-	-	-
YOFO [32]	2022	48.6	47.5	49.7	53.3	48.8	57.8
LBDT [18]	2022	49.4	48.2	50.6	54.3	-	-
MLSA [62]	2022	49.7	48.4	50.9	57.9	53.8	62.0
PMINet + CFBI [19]	2021	54.2	53.0	55.5	-	-	-
MTTR [8]	2022	55.3	54.0	56.6	-	-	-
CITD [37]	2021	61.4	60.0	62.7	-	-	-
ReferFormer [64]	2022	62.9	61.3	64.6	61.1	58.1	64.1
R^2-VOS [35]	2023	61.3	59.6	63.1	-	-	-
SgMg [49]	2023	**65.7**	**63.9**	**67.4**	63.3	60.6	66.0
OnlineRefer [63]	2023	63.5	61.6	65.5	64.8	61.6	67.7
LLM-based methods with reasoning ability							
LISA-7B [31]	2023	50.2	49.7	50.6	58.4	54.9	61.9
LISA-13B [31]	2023	52.6	52.1	53.0	60.7	56.8	64.6
TrackGPT-7B [75]	2023	56.4	55.3	57.4	63.2	59.4	67.0
TrackGPT-13B [75]	2023	59.5	58.1	60.8	66.5	62.7	70.4
PixelLM-7B [58]	2024	52.9	54.0	51.9	56.6	54.5	58.7
VideoLISA-3.8B (One-Token-Seg-All)	2024	61.7	60.2	63.3	<u>67.7</u>	<u>63.8</u>	<u>71.5</u>
VideoLISA-3.8B (Post-optimization)	2024	<u>63.7</u>	<u>61.7</u>	<u>65.7</u>	**68.8**	**64.9**	**72.7**

LoRA [25] and train the mask decoder with full finetuning. We train our model using 64 NVIDIA 24G A10 GPUs with a distributed training script based on DeepSpeed [56]. We use the AdamW [44] optimizer with the learning rate and weight decay set to 0.0003 and 0, respectively. We also adopt WarmupDecayLR as the learning rate scheduler, with the warmup iterations set to 100. The weights of the text generation loss (λ_{txt}) and the mask loss (λ_{seg}) are both set to 1.0. The weights of the BCE loss (λ_{bce}) and the DICE loss (λ_{dice}) are set to 2.0 and 0.5, respectively. The per-device batch size is set to 2. For ablation studies, the total number of iterations is $3,000$ and each experiment takes around 10 hours. For the final model used for comparison, we scale up the training to $6,000$ iterations, which takes 20 hours.

Evaluation Metrics For image-based evaluation, we adopt two metrics commonly used in previous works [28, 31]: gIoU and cIoU. gIoU is defined by the average of all per-image Intersection-over-Unions (IoUs), while cIoU is defined by the cumulative intersection over the cumulative union. For video-based evaluation, we follow previous practices [64, 63] and use region similarity (J), contour accuracy (F), and their average value (J&F).

5.2 Evaluation on Video Tasks

5.2.1 Referring Video Object Segmentation

We adopt two benchmarks of standard referring video object segmentation. Ref-Youtube-VOS is evaluated on the official challenge server [2]. Ref-DAVIS-17 is evaluated by the official evaluation code [3]. The evaluation results are shown in Tab. 1. Our method demonstrates competitive performance on both benchmarks, achieving comparable or superior results to existing methods. For Refer-DAVIS-17, our method achieves state-of-the-art performance, outperforming all the other methods by a considerable margin. In Refer-YouTube-VOS, our method performs well compared to traditional RVOS methods, achieving a high rank. State-of-the-art methods, such as SgMg [49], achieve remarkable performance, thanks to its dedicated video backbones, such as Video-Swin [43]. However, among LLM-based methods with reasoning ability, our model, despite having only 3.8B parameters, outperforms other methods with much larger LLMs, such as LISA-13B and TrackGPT-13B.

5.2.2 Motion-guided Video Object Segmentation

We further evaluate our model on motion-guided VOS using the MeViS [14] benchmark. Consistent with previous studies [14, 24], we evaluate our model's performance on the validation set of the MeViS benchmark. The results in Tab. 2 demonstrate that our method achieves state-of-the-art performance in this benchmark, outperforming previous methods by a large margin. We attribute this performance

---

[2]https://codalab.lisn.upsaclay.fr/competitions/3282

[3]https://github.com/davisvideochallenge/davis2017-evaluation

7

Table 2: Results on MeViS benchmark.

Methods	Year	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
URVOS [59]	2020	27.8	25.7	29.9
LBDT [18]	2022	29.3	27.8	30.8
MTTR [8]	2022	30.0	28.8	31.2
ReferFormer [64]	2022	31.0	29.8	32.2
VLT+TC [16]	2021	35.5	33.6	37.3
LMPM [14]	2023	37.2	34.2	40.2
VideoLISA-3.8B (One-Token-Seg-All)	2024	42.3	39.4	45.2
VideoLISA-3.8B (Post-optimization)	2024	**44.4**	**41.3**	**47.6**

Table 3: Results on ReasonVOS benchmark.

Methods	Year	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
MTTR [8]	2022	31.1	29.1	33.1
ReferFormer [64]	2022	32.9	30.2	35.6
SOC [46]	2023	35.9	33.3	38.5
OnlineRefer [63]	2023	38.7	34.6	42.9
SgMg [49]	2023	36.2	33.7	38.7
LISA [31]	2023	31.1	29.1	33.1
PixelLM [58]	2024	32.2	30.8	33.7
VideoLISA-3.8B (One-Token-Seg-All)	2024	45.1	43.1	47.1
VideoLISA-3.8B (Post-optimization)	2024	**47.5**	**45.1**	**49.9**

Table 4: Reasoning segmentation results among ours and previous related works. 'ft' denotes using 239 reasoning segmentation image-instruction pairs to finetune the model.

Method	val overall		test short query		test long query		test overall	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
OVSeg [38]	28.5	18.6	18.0	15.5	28.7	22.5	26.1	20.8
GRES [40]	22.4	19.9	17.6	15.0	22.6	23.8	21.3	22.0
X-Decoder [76]	22.6	17.9	20.4	11.6	22.2	17.5	21.7	16.3
SEEM [77]	25.5	21.2	20.1	11.5	25.6	20.8	24.3	18.7
Grounded-SAM [57]	26.0	14.5	17.8	10.8	22.4	18.6	21.3	16.4
LISA-7B [31]	44.4	46.0	37.6	34.4	36.6	34.7	36.8	34.1
LISA-7B (ft) [31]	52.9	54.0	40.6	40.6	49.4	51.0	47.3	48.4
LISA-13B [31]	48.9	46.9	39.9	**43.3**	46.4	46.5	44.8	45.8
LISA-13B (ft) [31]	56.2	62.9	**44.3**	42.0	54.0	54.3	51.7	51.1
VideoLISA-3.8B (Ours)	**61.4**	**67.1**	43.8	42.7	**56.9**	**57.7**	**53.8**	**54.4**

gap to our model's adeptness in capturing temporal dynamics and cross-modal interaction, facilitated by the Sparse Dense Sampling-based temporal module and the One-Token-Seg-All training paradigm.

5.2.3 Reasoning Video Object Segmentation

In Tab. 3, we compare various methods on the newly organized ReasonVOS benchmark. For traditional VOS methods, the metrics are evaluated using their released checkpoints pre-trained on the Ref-YouTube-VOS dataset. This benchmark focuses on complex reasoning, temporal understanding, and segmentation temporal consistency, which present significant challenges for existing VOS methods and image-based reasoning segmentation methods. It can be observed that most previous methods exhibit unsatisfactory performance on this benchmark. Traditional RVOS methods, such as ReferFormer [64], excel at tracking moving objects but struggle with comprehending complex language expressions, particularly those requiring multi-step reasoning with world knowledge. On the other hand, LLM-based models, like LISA [31] and PixelLM [58], have better language understanding and reasoning capabilities. The main reasons for the poor performance are: 1) incapability to capture temporal dynamics in the video, and 2) difficulty in segmenting temporally consistent masks. In contrast, our VideoLISA model demonstrates remarkable performance, thanks to the advanced model design that considers all these crucial aspects.

5.3 Evaluation on Image Tasks

We use the image reasoning segmentation benchmark [31] to assess the reasoning capability of our model. During testing, we duplicate an image into multiple frames as a pseudo video. The results are shown in Tab. 4. We observe that our VideoLISA achieves state-of-the-art performance on both validation set and test set. Remarkably, despite our model employing an LLM with significantly fewer parameters, it outperforms larger models, such as LISA-7B and LISA-13B, demonstrating its exceptional reasoning capability. We attribute the impressive performance to the following aspects. From a data perspective, VideoLISA benefits from joint training on both image and video datasets, allowing it to learn from more abundant and diverse supervision signals. On the model aspect, the temporal learning module and the One-Token-Seg-All training encourage the model to leverage multiple frames of video simultaneously to conduct reasoning, rather than focusing on one image. Even when generalizing to image tasks, where the video is simulated by an image, the model's reasoning capability remains effective. We provide more experiment results on image referring segmentation in the appendix. These experiments demonstrate that our model is capable of image-based tasks, suggesting the potential for unifying image/video referring/reasoning segmentation tasks into a language-instructed object segmentation task solvable by a single VideoLISA model.

8

Table 5: Ablation study on the temporal modeling architecture.

Method	ReasonSeg (val)		MeViS (valid_u)		
	giou	ciou	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
LISA-7B (Baseline)	51.7	56.7	43.2	39.9	46.5
LISA-7B (Vid. FT)	48.6	56.2	44.8	41.1	48.6
LLaMA-VID-7B [36] (Q-Former)	42.0	41.9	47.9	45.0	50.8
VideoLISA-3.8B (n-frame)	55.6	**60.8**	49.9	46.7	53.0
VideoLISA-3.8B (ST Pooling [47])	56.0	59.9	50.8	47.8	53.8
VideoLISA-3.8B (Slow-Fast Pooling [26])	54.0	54.4	50.2	47.2	53.1
VideoLISA-3.8B (Sparse Dense Sampling)	**58.9**	60.0	**51.7**	**48.4**	**54.9**

Table 6: Ablation study on the mask association *i.e.,* tracking architecture.

Method	MeViS (valid_u)		
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
LISA-7B (Baseline)	43.2	39.9	46.5
LISA-7B + XMem[12]	45.6	41.9	49.3
VideoLISA-3.8B (One-Token-Seg-One)	46.1	42.4	49.8
VideoLISA-3.8B (One-Token-Seg-All)	51.7	48.4	54.9
VideoLISA-3.8B (Post optimization)	**54.5**	**50.9**	**58.1**

5.4 Ablation Studies

We conduct ablation studies on various design choices of our model. The detailed experiment results are provided in the appendix. Here, we summarize the main takeaways for each study.

Ablation of temporal learning module. In this study of Tab. 5, we compare our Sparse Dense Sampling strategy with various design choices, including LISA [31] finetuned on videos, Q-Former [33] architecture inherited from LLaMA-VID [36], one straightforward solution that directly concatenate visual tokens from multiple frames (n-frame), a strategy that pools along spatial and temporal dimension separately (ST Pooling), a strategy that pools each frame with different strengths in a slow-fast pace. The comparison of the experiment results shows that our Sparse Dense Sampling strategy outperforms other video-LLM training (sampling) strategies. In addition to demonstrating the effectiveness of our method, this study also reveals the unique properties of the VOS task. On the one hand, it requires detailed visual information for accurate segmentation, which makes the pooling-based strategies yield inferior results. On the other hand, temporal information is also necessary for the model to comprehend motions and behaviors, as validated by the comparison between n-frame and ours.

Ablation of temporal association module. The main takeaway of this part, as shown in Table 5, lies in the comparison between our method and extensions of image-based LISA. Specifically, we upgrade LISA to fit the VOS task by 1) (baseline) using one <SEG> token from the first frame to segment subsequent frames, 2) marrying LISA with an off-the-shelf tracking model. With the help of the tracker, LISA performs clearly better than the baseline, while still performs worse than our method. The main issue comes from that without perception of the video, the model is incapable of processing queries that are concerned with the full video content and temporal dynamic. We further quantify the effect of the One-Token-Seg-All approach by contrasting it with a strawman setting, One-Token-Seg-One. The comparison clearly validates the effect and necessity of the One-Token-Seg-All approach.

6 Limitation and Future Work

Despite the remarkable performance shown on various benchmarks, our model still has limitations. We discuss them in this section to inspire future work. First, our model exhibits deficiencies in computational efficiency. Although we have already reduced the size of LLM to 3.8B, which is much smaller than previous models (7B, 13B), it still incurs a relatively high computational cost compared to previous work on video object segmentation. In other words, introducing a MLLM brings remarkable understanding and reasoning ability to the model, while also inducing computational costs. Exploring methods to achieve a trade-off between these aspects presents an interesting avenue for future research. Second, we observe that state-of-the-art approaches to video object segmentation often employ dedicated video backbones to enhance performance. Intuitively, using vision encoder pre-trained on videos would be beneficial for temporal-related tasks, such as object tracking. However, integrating a video backbone while ensuring compatibility with LLM and SAM decoder is non-trivial. In this work, we focus on empowering video segmentation tasks with reasoning capabilities based on LLM. Exploring the integration of a video backbone represents a potential avenue for future research. Lastly, after re-purposing the MLLM into an expert model on video reasoning segmentation, we observe that the model exhibits significant performance degradation in text generation capability, *i.e.,* chat. We notice that this is a common limitation among reasoning segmentation models. LISA model even shows a much worse performance. Developing segmentation capability while preserving the chat capability is not trivial as it involves various aspects, such as data curation, training strategy, etc.

9

7 Conclusion

In this work, we propose VideoLISA, a video-based LLM designed for language instructed reasoning segmentation in videos. It leverages the reasoning capabilities of LLM and employs SAM to produce segmentation masks. To address the unique challenges in marrying LLM with video object segmentation, we propose two key innovations. Firstly, a Sparse Dense Sampling strategy is designed to enable LLM to capture and understand temporal dynamics in videos. By leveraging the inherent temporal redundancy property of videos, this strategy achieves a delicate balance between preserving visual details and temporal context, making it favorable for video object segmentation tasks. Secondly, we propose a One-Token-Seg-All approach to achieve temporally consistent segmentation masks in the promptable mask decoding paradigm. Based on a dedicated investigation of the potential and challenges associated with using a single unified prompt to segment video frames, we enhance this capability from both input information foundation and training objective perspectives. Extensive ablation studies have investigated the function and rationale of the design choices of two modules. Equipped with the two designs above, our VideoLISA model shows impressive capabilities in video object segmentation, particularly emphasizing complex reasoning, temporal understanding, and object tracking, as validated by our newly organized ReasonVOS benchmark. Furthermore, it demonstrates notable performance on image segmentation tasks, positioning it as a potential unified model for language-instructed object segmentation.

Acknowledgement

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG3-RP-2022-030).

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[2] Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan. Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1674–1683, 2023.

[3] Zechen Bai, Yuta Nakashima, and Noa Garcia. Explain me the painting: Multi-topic knowledgeable art description generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5422–5432, 2021.

[4] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

[5] Maksym Bekuzarov, Ariana Bermudez, Joon-Young Lee, and Hao Li. Xmem++: Production-level video segmentation from few annotated frames. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 635–644, 2023.

[6] Miriam Bellver, Carles Ventura, Carina Silberer, Ioannis Kazakos, Jordi Torres, and Xavier Giro-i Nieto. A closer look at referring expressions for video object segmentation. *Multimedia Tools and Applications*, 82(3):4419–4438, 2023.

[7] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pages 850–865. Springer, 2016.

[8] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4985–4995, 2022.

[9] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[10] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

10

[11] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014.

[12] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022.

[13] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2694–2703, 2023.

[15] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20224–20234, 2023.

[16] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16321–16330, 2021.

[17] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16321–16330, 2021.

[18] Zihan Ding, Tianrui Hui, Junshi Huang, Xiaoming Wei, Jizhong Han, and Si Liu. Language-bridged spatial-temporal interaction for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4964–4973, 2022.

[19] Zihan Ding, Tianrui Hui, Shaofei Huang, Si Liu, Xuan Luo, Junshi Huang, and Xiaoming Wei. Progressive multimodal interaction network for referring video object segmentation. *The 3rd Large-scale Video Object Segmentation Challenge*, 8:6, 2021.

[20] Ke Fan, Zechen Bai, Tianjun Xiao, Tong He, Max Horn, Yanwei Fu, Francesco Locatello, and Zheng Zhang. Adaptive slot attention: Object discovery with dynamic slot number. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23062–23071, 2024.

[21] Ke Fan, Zechen Bai, Tianjun Xiao, Dominik Zietlow, Max Horn, Zixu Zhao, Carl-Johann Simon-Gabriel, Mike Zheng Shou, Francesco Locatello, Bernt Schiele, et al. Unsupervised open-vocabulary object localization in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13747–13755, 2023.

[22] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[23] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5958–5966, 2018.

[24] Shuting He and Henghui Ding. Decoupling static and hierarchical motion perception for referring video segmentation. *arXiv preprint arXiv:2404.03645*, 2024.

[25] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[26] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. *arXiv preprint arXiv:2403.19046*, 2024.

[27] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv preprint arXiv:2311.08046*, 2023.

11

[28] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[29] 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.

[30] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[31] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

[32] Dezhuang Li, Ruoqi Li, Lijun Wang, Yifan Wang, Jinqing Qi, Lu Zhang, Ting Liu, Qingquan Xu, and Huchuan Lu. You only infer once: Cross-modal meta-transfer for referring video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1297–1305, 2022.

[33] 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.

[34] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[35] Xiang Li, Jinglu Wang, Xiaohao Xu, Xiao Li, Bhiksha Raj, and Yan Lu. Robust referring video object segmentation with cyclic structural consensus. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22236–22245, 2023.

[36] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2025.

[37] Chen Liang, Yu Wu, Tianfei Zhou, Wenguan Wang, Zongxin Yang, Yunchao Wei, and Yi Yang. Rethinking cross-modal interaction from a top-down perspective for referring video object segmentation. *arXiv preprint arXiv:2106.01061*, 2021.

[38] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, 2023.

[39] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

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

[41] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[42] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4761–4775, 2021.

[43] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

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

[45] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 10034–10043, 2020.

[46] Zhuoyan Luo, Yicheng Xiao, Yong Liu, Shuyan Li, Yitong Wang, Yansong Tang, Xiu Li, and Yujiu Yang. Soc: Semantic-assisted object cluster for referring video object segmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

12

[47] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[48] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[49] Bo Miao, Mohammed Bennamoun, Yongsheng Gao, and Ajmal Mian. Spectrum-guided multi-granularity referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 920–930, 2023.

[50] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21033–21043, 2022.

[51] Shehan Munasinghe, Rusiru Thushara, Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, Mubarak Shah, and Fahad Khan. Pg-video-llava: Pixel grounding large video-language models. *arXiv preprint arXiv:2311.13435*, 2023.

[52] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[53] Renjie Pi, Jiahui Gao, Shizhe Diao, Rui Pan, Hanze Dong, Jipeng Zhang, Lewei Yao, Jianhua Han, Hang Xu, and Lingpeng Kong Tong Zhang. Detgpt: Detect what you need via reasoning. *arXiv preprint arXiv:2305.14167*, 2023.

[54] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023.

[55] Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad S. Khan. Llava++: Extending visual capabilities with llama-3 and phi-3, 2024.

[56] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.

[57] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[58] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024.

[59] 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.

[60] Li Wang, Zechen Bai, Yonghua Zhang, and Hongtao Lu. Show, recall, and tell: Image captioning with recall mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12176–12183, 2020.

[61] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11686–11695, 2022.

[62] Dongming Wu, Xingping Dong, Ling Shao, and Jianbing Shen. Multi-level representation learning with semantic alignment for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4996–5005, 2022.

[63] Dongming Wu, Tiancai Wang, Yuang Zhang, Xiangyu Zhang, and Jianbing Shen. Onlinerefer: A simple online baseline for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2761–2770, 2023.

13

[64] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4984, 2022.

[65] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024.

[66] Linjie Yang, Yuchen Fan, and Ning Xu. The 2nd large-scale video object segmentation challenge - video object segmentation track, October 2019.

[67] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18155–18165, 2022.

[68] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. Video object segmentation and tracking: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(4):1–47, 2020.

[69] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10502–10511, 2019.

[70] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

[71] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[72] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023.

[73] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[74] 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.

[75] Jiawen Zhu, Zhi-Qi Cheng, Jun-Yan He, Chenyang Li, Bin Luo, Huchuan Lu, Yifeng Geng, and Xuansong Xie. Tracking with human-intent reasoning. *arXiv preprint arXiv:2312.17448*, 2023.

[76] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *CVPR*, 2023.

[77] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv:2304.06718*, 2023.

14

A Appendix

A.1 Evaluation on Image Segmentation

In this section, we evaluate our VideoLISA model on the referring image segmentation task with three widely adopted benchmarks. The results are presented in Tab. 7. On the refCOCO and refCOCO+ benchmarks, our VideoLISA achieves comparable performance with the image-based LISA model. On the refCOCOg benchmark, VideoLISA outperforms previous methods, achieving state-of-the-art performance. In general, the results of this experiment, along with the image reasoning segmentation results shown in the main paper, effectively demonstrate that our VideoLISA model is a strong competitor in image segmentation tasks.

Table 7: Referring segmentation results (cIoU) among ours and existing methods.

Method	refCOCO			refCOCO+			refCOCOg	
	val	testA	testB	val	testA	testB	val(U)	test(U)
MCN [45]	62.4	64.2	59.7	50.6	55.0	44.7	49.2	49.4
VLT [17]	67.5	70.5	65.2	56.3	61.0	50.1	55.0	57.7
CRIS [61]	70.5	73.2	66.1	62.3	68.1	53.7	59.9	60.4
LAVT [67]	72.7	75.8	68.8	62.1	68.4	55.1	61.2	62.1
ReLA [40]	73.8	76.5	70.2	**66.0**	**71.0**	**57.7**	65.0	66.0
X-Decoder [76]	-	-	-	-	-	-	64.6	-
SEEM [77]	-	-	-	-	-	-	65.7	-
LISA-7B [31]	**74.1**	76.5	**71.1**	62.4	67.4	56.5	66.4	68.5
VideoLISA-3.8B (Ours)	73.8	**76.6**	68.8	63.4	68.8	56.2	**68.3**	**68.8**

A.2 Ablation Studies

In this section, we present ablation studies on the temporal learning module (the Sparse Dense Sampling strategy), the temporal mask association module (the One-Token-Seg-All approach), and the training data recipe. For fair comparisons, unless specified, all VideoLISA variants are uniformly trained with the same training setting: 1) 3k iterations in total, 2) the same training data recipe, 3) the same learning rate scheduler, and 4) the same training objective. Three benchmarks are used for analysis: 1) ReasonSeg [31] evaluates the reasoning ability of the model; 2) MeViS [14] reflects the model's performance on temporal learning; and 3) Ref-DAVIS-17 [29] measures the general RVOS capability of the model. For evaluation on video benchmarks, the performance metrics of VideoLISA are computed using the simple One-Token-Seg-All approach without post-optimization, revealing the model's essential capabilities.

A.2.1 Temporal Learning Module

In Tab 8, we compare various strategies for temporal learning. The first row shows the vanilla LISA-7B model, which only focuses on image-based reasoning segmentation. To infer LISA-7B on video data, we employ a similar One-Token-Seg-All strategy, where the <TRK> token (called [SEG] in the original LISA) comes from the first frame. This performance serves as a baseline for comparison. In the second row, we construct a naive solution to adapt LISA to the video domain. Specifically, we finetune LISA-7B on the aforementioned video segmentation datasets. The results show that simply finetuning on video data does not significantly improve video performance and even hurts the performance on image reasoning segmentation. Although training on video datasets may enhance the model's ability to understand temporally related text queries, it still lacks temporal modeling ability from video data, resulting in undesirable performance.

Next, we experiment with the classical Q-Former [33] architecture using LLaMA-VID [36] implementation thanks to its exceptional performance on video understanding. Specifically, LLaMA-VID compresses each video frame into two tokens using the Q-Former architecture, reducing computational cost. We adopt LLaMA-VID and add a segmentation head similar to VideoLISA. We observe that LLaMA-VID, equipped with the segmentation head, is capable of doing segmentation and achieves decent performance across benchmarks, especially in video benchmarks. When being compared with VideoLISA, LLaMA-VID shows worse performance across all evaluated benchmarks.

15

Table 8: Ablation study on the temporal modeling architecture. *LISA-7B is reproduced using the released codebase.

Method	ReasonSeg (val)		MeViS (valid_u)			Ref-DAVIS-17		
	giou	ciou	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
LISA-7B* (Baseline)	51.7	56.7	43.2	39.9	46.5	58.8	55.1	62.5
LISA-7B* (Vid. FT)	48.6	56.2	44.8	41.1	48.6	58.5	54.6	62.5
LLaMA-VID [36] (Q-Former)	42.0	41.9	47.9	45.0	50.8	62.1	58.9	65.4
VideoLISA-3.8B (n-frame)	55.6	**60.8**	49.9	46.7	53.0	65.5	62.2	68.9
VideoLISA-3.8B (Spatial & Temporal Pooling [47])	56.0	59.9	50.8	47.8	53.8	62.2	58.4	66.3
VideoLISA-3.8B (Slow-Fast Pooling [26])	54.0	54.4	50.2	47.2	53.1	65.7	62.1	69.4
VideoLISA-3.8B (Sparse Dense Sampling)	**58.9**	60.0	**51.7**	**48.4**	**54.9**	**67.8**	**64.3**	**71.3**

Table 9: Ablation study on the mask association *i.e.,* tracking architecture. *LISA-7B is reproduced using the released codebase.

Method	MeViS (valid_u)			Ref-DAVIS-17		
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
LISA-7B* (Baseline)	43.2	39.9	46.5	58.8	55.1	62.5
LISA-7B* + XMem[12]	45.6	41.9	49.3	62.7	60.0	65.5
VideoLISA-3.8B (One-Token-Seg-One)	46.1	42.4	49.8	60.2	56.5	63.8
VideoLISA-3.8B (One-Token-Seg-All)	51.7	48.4	54.9	67.8	64.3	71.3
VideoLISA-3.8B (Post optimization)	**54.5**	**50.9**	**58.1**	**68.7**	**65.5**	**72.0**

The performance gap comes from that LLaMA-VID compressing the visual tokens into extremely low resolution, i.e., two tokens only. This compression inevitably lost visual spatial details, which are essential for segmentation.

After that, we compare various temporal learning strategies within the VideoLISA framework using the One-Token-Seg-All training objective. We first experiment with a straightforward video training strategy, called n-frame, which directly concatenates the visual features from n sampled frames as input to the large language model. In our implementation, the value of n is set to the same as T_{dense} for comparison. As shown in the third row, we observe that with this simple strategy, the model achieves surprisingly good performance across the benchmarks, significantly outperforming LISA-based methods. Exposure to multiple frames enables the model to perceive temporal dynamics, while the One-Token-Seg-All training objective supervises the model in learning mask association over the temporal dimension, thereby improving multimodal reasoning and temporal consistency in segmentation. However, due to computational limits, it is prohibitive to include too many frames as it would result in a large number of tokens.

To enable long temporal context perception, we experiment with several pooling strategies, including pooling along the spatial and temporal dimensions separately [47], pooling with different strengths in a slow-fast pace [22, 26], and our Sparse Dense Sampling strategy. The comparison in Tab. 8 reveals that our Sparse Dense Sampling strategy is a more favorable setting among the experiment designs. The first spatial-temporal pooling strategy eliminates valuable visual details of the video, resulting in inferior performance. The second slow-fast paced pooling strategy is similar to ours in implementation. The key difference is that it applies pooling to all frames, albeit with different strengths, while ours preserves the full visual details of the dense frames. This difference leads to the observed performance gap. We argue that this difference is significant due to the unique nature of the video object segmentation task. On one hand, it requires detailed visual information for accurate segmentation, causing pooling-based strategies to yield inferior results. On the other hand, the temporal dimension is also necessary for the model to comprehend motions and behaviors, as validated by the comparison between the n-frame approach and ours. Although recent studies [65] show that applying pooling to visual tokens does not affect the performance of VQA tasks, our experiments validate that preserving the full resolution of visual tokens is necessary for dense prediction tasks, and applying pooling leads to sub-optimal results.

A.2.2 Temporal Association Module

In Tab. 9, we compare the design choices for the temporal association module, *i.e.,* tracking. As in previous comparisons, the One-Token-Seg-All strategy in LISA-7B serves as the baseline in the first row. One straightforward solution based on LISA is to plug an off-the-shelf tracker into the

16

Table 10: Ablation study on the training data recipe.

Training Data				ReasonSeg (val)		MeViS (valid_u)			Ref-DAVIS-17		
Image Seg.	Video Seg.	Image QA	Video QA	giou	ciou	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
✓				57.2	60.0	46.0	43.3	48.6	62.6	58.9	66.3
	✓			41.4	46.5	49.3	45.8	52.8	66.0	62.7	69.3
✓	✓			58.9	60.0	51.7	48.4	54.9	**67.8**	**64.3**	**71.3**
✓	✓	✓		56.0	65.6	49.8	46.8	52.9	66.8	63.4	70.3
✓	✓	✓	✓	**60.6**	**67.4**	**52.0**	**49.3**	54.8	66.9	63.5	70.3

model. During inference, LISA outputs the segmentation mask of the first frame based on language instruction. The tracker then tracks the segmented object through the video, yielding segmentation masks for the subsequent frames. Specifically, we adopt the popular XMem [12] model as the tracker, as shown in the second row of the table. Compared to VideoLISA (both One-Token-Seg-All and post-optimization), LISA+XMem achieves worse performance on these benchmarks. This validates that simply plugging an existing tracker into an image-based reasoning segmentation model does not address the problem of video reasoning segmentation. The vital issue is that the LLM in charge of perception and reasoning does not capture the entire video content, making its predictions nonsensical. In contrast, VideoLISA's temporal learning module and dedicated training objective enrich the <TRK> token with semantic information, enabling it to find the target object across all frames.

To quantify the effect of the One-Token-Seg-All training objective, we build a strawman setting named One-Token-Seg-One. In this setting, the video content is captured with the temporal learning module, but the training only supervises the segmentation of one frame. The comparison is shown in the third and fourth rows of Tab. 9. We observe that the slight difference in supervision leads to a significant performance gap in the benchmarks. This indicates that the One-Token-Seg-All training objective is essential for achieving temporally consistent masks.

In the last row, we present post-optimization, which leverages both the reasoning and segmentation abilities of VideoLISA and a mature tracking model. Specifically, we first use VideoLISA to produce the <TRK> token and then use it to segment the sampled dense frames. Then, the post-optimization model, implemented as XMem++ [5], takes dense frames and their segmentation masks as references in its permanent memory and infers the masks for the remaining frames. The reasons for choosing the dense frames as the mask reference include: 1) the dense frames are seen by VideoLISA, thus their masks should be more accurate than those of other unseen frames, and 2) the dense frames are intentionally sampled from the video in a uniform manner, naturally providing a long-range yet diverse reference signal. By leveraging the association ability from the post-optimization step, VideoLISA achieves the best performance.

A.2.3 Ablation on Training Data

Our model undergoes joint training on both image and video datasets. An investigation of the training data is presented in Tab. 10. We first observe that with image-only segmentation datasets, the model achieves decent performance in reasoning segmentation. However, the performance on video benchmarks is unsatisfactory, possibly due to insufficient temporal information in the training data. When using video-only segmentation settings, compared to image-only, the performance on video benchmarks increases significantly. Simultaneously, the model experiences a dramatic drop in performance in reasoning segmentation. This comparison demonstrates that video training is helpful for the VOS task, while image data is also necessary to exploit the reasoning ability of the model. When combining the image and video segmentation datasets, the model yields remarkable performance across various benchmarks.

Next, we additionally explore the effect of using visual question answering (VQA) data. We first observe that after adding Image-QA data into training, the model experiences a slight performance drop in all benchmarks. Then, with the involvement of Video-QA data, the model achieves much better performance on the reasoning segmentation benchmark. Among the two video benchmarks, compared to the model trained with segmentation-only data, this model shows slightly better performance on the MeViS offline validation set yet worse performance on Ref-DAVIS-17. Intuitively, VQA data has the potential to enhance the model's reasoning ability. However, it may also make the multi-task training more challenging, as revealed by the performance fluctuation among different benchmarks. Maintaining the compatibility of different types of training data and tasks is left for future work.

17

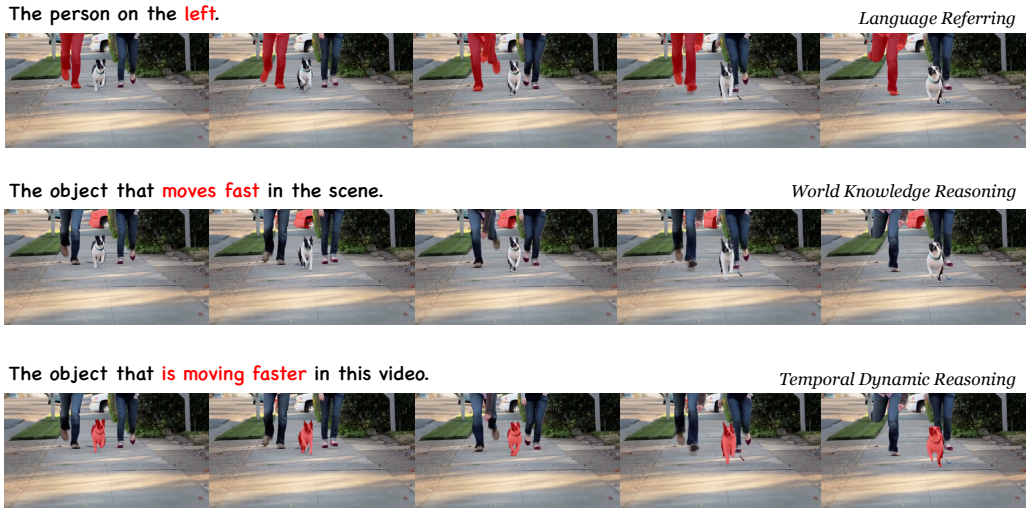



Figure 3: VideoLISA is a capable model on video object segmentation with versatile language-instructed reasoning abilities. Beyond basic language referring, it enables complex reasoning by leveraging world knowledge and videos temporal dynamics.

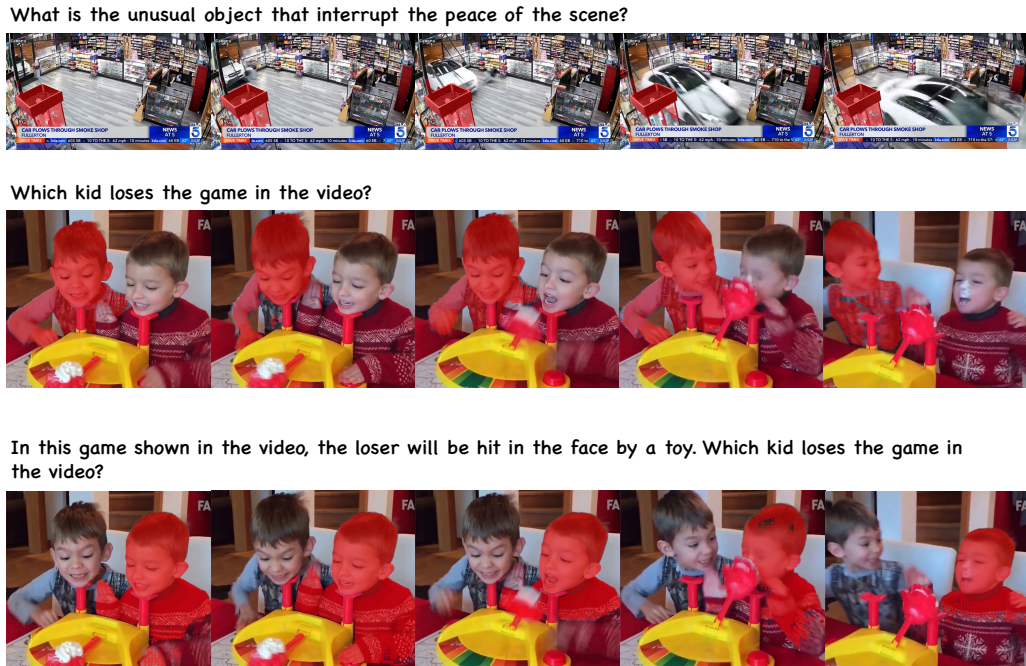



Figure 4: Failure cases of VideoLISA.

A.3 Qualitative Results

In Fig. 3, we use a representative video to showcase the versatile language-instructed reasoning capabilities of our model. VideoLISA can do segmentation in videos via language referring, world knowledge reasoning, and video temporal reasoning. Additionally, the model can discern subtle differences in language instructions and is not biased to salient or moving objects.

In Fig. 6 and Fig. 7, we provide more abundant qualitative examples of VideoLISA. The red text is only for illustration purposes. No special prompting techniques were employed. It's important to note that these examples were generated using the One-Token-Seg-All inference approach without post-optimization.

18

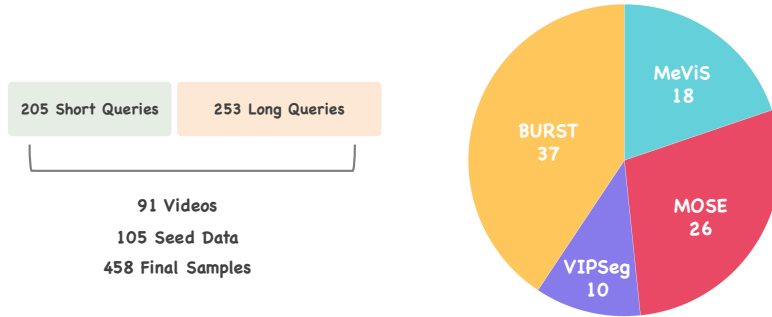



Figure 5: ReasonVOS benchmark. The left part shows the statistics of data samples. The right part shows the source of the videos.

A.4 Failure Cases

To understand the limitations and capability boundaries of our method, we analyze several failure cases as shown in Fig. 4. In the first example, the video shows a car crashing into a grocery store. We prompt the model to find the unusual object that interrupts the peace of the scene. Although we try to rephrase the prompt in various ways, the model consistently outputs the object in the bottom left corner. We hypothesize that the issue stems from the inherent hallucination of the MLLM, which recognizes the object as a stove, a telephone pole, or something else.

In the second example, we ask the model to find the kid who loses the game We humans have the background knowledge to determine the match result. However, it seems like this game is beyond the knowledge scope of the MLLM, causing it to segment the wrong person. Consequently, we provide some background information about the game rules in the text prompt and then ask the same question. As shown in the third example of Fig. 4, with this cue, the model is able to segment the correct person. These examples demonstrate that the reasoning capabilities of VideoLISA are bounded by the multimodal large language model behind it, yet this can be alleviated by prompt engineering techniques. The third example also exhibits low-quality segmentation masks in certain frames, leaving room for future improvements.

B Benchmark

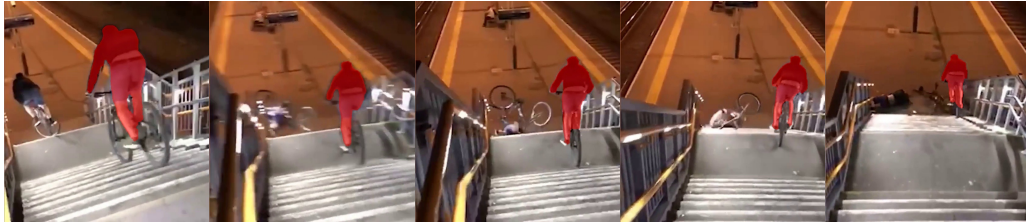
We show the data statistics of our ReasonVOS benchmark in Fig. 5. We select videos and mask annotations from various sources and annotate additional text descriptions. In total, ReasonVOS consists of 91 videos. We manually annotate 105 video-instruction-mask samples as seed data and use Claude 3 API to augment the data into 458 samples. We further categorize the text descriptions into short query and long query. Short queries are descriptions of specific objects, usually in the format of attributive clauses. Long queries are instructions that require reasoning, usually in the format of a full sentence.

C Broader Impact

The development of our reasoning-based video segmentation model holds significant potential for transforming a variety of fields by enhancing the ability to analyze and interpret video content. In the realm of surveillance, this technology can improve security measures by accurately identifying and tracking suspicious behavior, thereby preventing potential threats. In educational settings, the model can assist teachers in identifying and addressing student engagement patterns, fostering a more responsive learning environment. For healthcare, our model can be applied to monitor patient activities, supporting early intervention and personalized care strategies. Additionally, in everyday scenarios, such as pet care or home organization, this technology can assist individuals in making informed decisions quickly and efficiently. By leveraging advanced reasoning capabilities, our model not only advances the field of computer vision but also provides practical solutions that enhance safety, learning, health, and daily life. However, it is crucial to consider ethical implications, such as privacy concerns and the potential for misuse, ensuring that these technologies are implemented responsibly and equitably.

19

After the badminton game, who might be the guy that needs to go to the hospital to get a head injury checked out?

If I want to learn bicycle skill from one guy from this video, considering their bicycle skill and performance, which one should I turn to?

In this video, there is something that shocks the cat and makes it jump. Can you find the object?

I came home in the evening and asked my daughter how her day was. She told me she played tennis in the kitchen with her friend and had a great day. Which one in the video is the friend she mentioned?

Of the people in the scene, which one is more likely to be carrying food and drink? Please find him.

The construction site has halted work due to a shortage of materials. Which object is most likely being awaited?

Figure 6: More qualitative examples of VideoLISA.

20

In the Nurburgring race track, this vehicle has never been banned, but it is unusual to see it at this race. Can you find it?

Observing the group dynamics, who appears to be the lead singer taking the spotlight?

According to the scene, which object would be suitable to play throw and catch games? Please output the segmentation mask.

This dog seems to be trying to control something. Please segment the object that interests the dog.

The mother that leads a group of children ahead.

The object falling from the air and being caught by the dog.

Figure 7: More qualitative examples of VideoLISA.

21

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The method and experiments clearly support the claim made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have a dedicated section (Sec. 6) discussing the limitation and potential future works.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

22

Answer: [NA]

Justification: This paper mainly focus on model architecture design and experimental verification.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide sufficient details about the data and implementation in Sec. 5.1. We will make the code public upon publishing.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

23

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The data and code are temporarily not open access the submission period. We will make the data and code public upon publishing.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The implementation details are provided in Sec. 5.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Due to the demanding computational cost in our experiments, we currently don't report error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

24

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We elaborate the compute resource and time usage in Sec. 5.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We comply with the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discussed the broader impacts in Sec. C.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

25

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cited the data and code used in our paper and we comply with the license.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

26

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: In this work, we have made annotations for existing public dataset. We provide sufficient details for this process in Sec. 4.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

27