VIDEOPLUM: A LARGE MULTIMODAL MODEL FOR PART-LEVEL VISUAL GROUNDING IN VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Objects are composed of fine-grained parts, and identifying these parts can enable more precise visual reasoning and grounded language generation. Extending such fine-grained grounding to videos, however, requires identifying and consistently tracking object parts over time. Existing image-based segmentation and grounding methods do not readily address this temporal dimension at the part-level. We introduce VIDEOPLUM, a large multimodal model for fine-grained, part-level visual grounding in videos. Our approach combines the semantic reasoning capabilities of a vision-language model with SAM-based segmentation to generate accurate spatiotemporal masks for object parts, proposing an end-to-end approach for fine-grained, part-level grounding across video frames. To address the scarcity of fine-grained part-level grounding annotations in videos, we develop an automatic data-generation pipeline that identifies relevant object parts and temporally propagates their masks. We evaluate our model on both part-level and object-level grounding across videos and images. VIDEOPLUM substantially outperforms existing baselines in fine-grained part grounding, achieving 51.3 $\mathcal{J}\&\mathcal{F}$ across SMITE, PUMaVOS, and EgoFun3D and surpassing GLaMM + SAM2 by 12.5 points. Our method also generalizes comparably to image segmentation, extending its fine-grained visual reasoning capabilities beyond video grounding.

1 INTRODUCTION

Objects are composed of semantic parts whose form and function help distinguish one object from another. Recognizing and localizing these parts supports fine-grained visual understanding and reasoning (He et al., 2021; Felzenszwalb et al., 2010; Ramanathan et al., 2023; Blume et al., 2025). Extending this capability from images to video introduces a distinct challenge. A requested part must be localized within the intended object instance and remain associated with that instance as its appearance and visibility change over time. Motion, occlusion, and viewpoint variation can all weaken this object-part association. Semantic-part grounding in video must therefore model not only where a part appears, but also which object it belongs to throughout the sequence.

Recent work has given multimodal language models the ability to produce pixel masks from language in images (Lai et al., 2023; Rasheed et al., 2024; Wang et al., 2024; Ren et al., 2023). Yet, a recent work on image-level part segmentation, PARTONOMY (Blume et al., 2025), shows that grounding becomes substantially more difficult at the level of semantic parts. PLUM addresses part grounding in static images by selecting spans from ordinary generated text and feeding predicted masks back into later predictions. Video grounding adds temporal propagation, but usually is more focused on the grounding and segmentation of parent objects rather than small parts (Munasinghe et al., 2024; Bai et al., 2024; Cuttano et al., 2025; Yan et al., 2024; Yuan et al., 2025). Therefore we introduce VIDEOPLUM to cross the gap between object- and part-level grounding in video contexts.

VIDEOPLUM is an end-to-end video grounding and segmentation large multimodal model (LMM) built on Qwen3.5-4B (Qwen Team, 2026) and SAM2 (Ravi et al., 2024). Prior segmentation LMMs such as VISA (Yan et al., 2024), VideoGLaMM (Munasinghe et al., 2024), and Sa2VA (Yuan et al., 2025) extend the vocabulary with dedicated segmentation tokens e.g., [SEG] and decode masks from their embeddings, this makes the model rely on tokens unseen during pretraining which may cause a distribution shift. VideoPLUM instead keeps the LMM's output in ordinary text. Given a video and a question about an object part, Qwen answers in natural language, and a lightweight token classifier

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

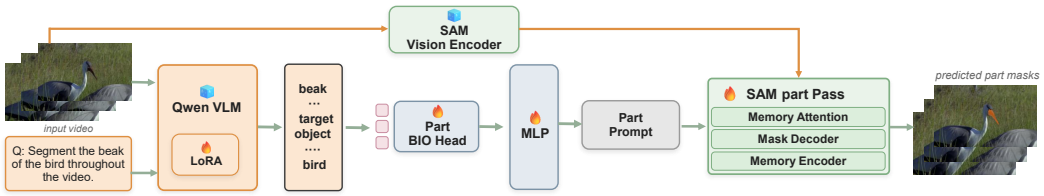


Figure 1: **Overview of VIDEOPLUM.** A BIO head identifies the target-part span in ordinary generated text. Its pooled hidden representation prompts SAM-2, whose temporal memory propagates the resulting part mask across the video.

tags each generated token with a BIO label (beginning, inside or outside) to mark the span that names the requested part. The hidden states of this span are pooled and passed to SAM2 through a text-prompt input that we add to its prompt encoder, so the part is segmented directly from the words that describe it, without the need to first segment the parent object. SAM2's temporal memory then carries the part's predictions from frame to frame, producing a consistent mask track across the video. The main intuition underlying this design is to leverage the backbone LMM's parametric knowledge of object-part relations and to evaluate whether this knowledge transfers to pixel-level part segmentation across the temporal dimension of videos.

Because dense semantic-part annotations are scarce in video, we construct part-mask supervision from MeViS, Refer-YouTube-VOS and its GCG variant, and VidSTG (Ding et al., 2023; Seo et al., 2020a; Munasinghe et al., 2024; Zhang et al., 2020). Stage 1 uses Qwen3.5-4B to propose semantic parts, LocateAnything (Wang et al., 2026) to localize them within parent objects, and SAM2 to generate and propagate masks. This yields a pseudo-labeled corpus of 8,563 parent-object tracks and 29,034 part requests. Stage 2 couples Qwen3.5-4B with the SAM2 video segmenter. We adapt Qwen using LoRA (Hu et al., 2021) and train it to mark part spans with BIO labels; the projected spans prompt SAM2 to decode masks and propagate them using temporal memory. VIDEOPLUM achieves 61.91, 56.55, and 35.40 $\mathcal{J}\&\mathcal{F}$ on SMITE, PUMaVOS, and EgoFun3D, respectively, compared with 31.59, 49.32, and 35.46 for GLaMM + SAM2, the strongest baseline on average. Its mean score is 51.29, exceeding the baseline's 38.79 by 12.50 points.

Contributions.

1. We study language-guided semantic-part grounding in video, which requires a requested part to be localized and tracked from natural-language descriptions.
2. We introduce VIDEOPLUM, which identifies part-bearing spans in ordinary generated text and projects their representations into SAM-2 for temporally consistent mask decoding, without requiring dedicated segmentation tokens.
3. We construct temporal semantic-part supervision from existing object-level video datasets. VIDEOPLUM achieves an average $\mathcal{J}\&\mathcal{F}$ score of 51.29, compared with 38.79 for the strongest baseline.

2 RELATED WORK

Segmentation-enabled multimodal models. Semantic-part segmentation decomposes objects into semantically meaningful regions. Datasets such as PASCAL-Part, ADE20K-Part, PartImageNet, and PACO provide pixel-level part annotations spanning different object categories and levels of granularity (Chen et al., 2014; Zhou et al., 2017; 2019; He et al., 2021; Ramanathan et al., 2023). Recent pixel-level segmentation work has adopted vision-language models (VLMs) for language-driven mask prediction. LISA (Lai et al., 2023) connects a multimodal language model to SAM through a dedicated <SEG> token, PixelLM (Ren et al., 2023) extends this framework to multiple segmentation outputs, and GLaMM (Rasheed et al., 2024) associates phrases in generated responses with masks. Although these models may be trained with part-level annotations, PARTONOMY (Blume et al., 2025) shows that even strong segmentation-enabled multimodal models struggle to ground object-specific parts and relate them to their parent objects. PARTONOMY also introduces

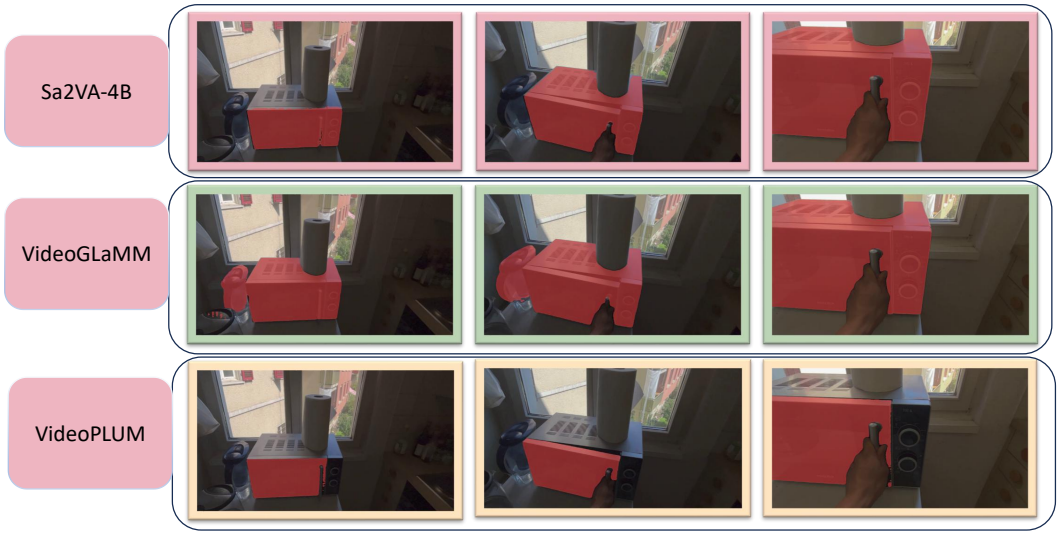


Figure 2: An example of VIDEOPLUM's Part-grounding capabilities compared to recent LMMs. The part is "Microwave door"

PLUM, which identifies mask-bearing spans by assigning BIO labels to ordinary generated tokens and uses predicted masks to inform subsequent mask predictions. VIDEOPLUM adopts this span-based interface for video and extends this span-based interface to video, using the parent-object expression to contextualize the requested part and SAM-2's memory to propagate its mask.

Language-guided video segmentation and grounding. Referring video object segmentation predicts a sequence of masks for an object specified by a natural-language expression. Benchmarks such as A2D-Sentences (Gavrilyuk et al., 2018), Ref-DAVIS17 (Khoreva et al., 2018), Ref-YouTube-VOS (Seo et al., 2020a), and MeViS (Ding et al., 2023) have supported methods including URVOS, ReferFormer, OnlineRefer, and SgMg (Seo et al., 2020b; Wu et al., 2022; 2023; Miao et al., 2023). More recent multimodal models broaden the range of instructions that can specify the target. PG-Video-LLaVA combines generated descriptions with external detection and tracking, while Track-GPT (Zhu et al., 2023) introduces instruction-based tracking (Munasinghe et al., 2023). VideoLISA uses a single <TRK> token to prompt masks across frames, and VideoGLaMM associates generated phrases with spatiotemporal masks through <SEG> tokens and a video mask decoder (Bai et al., 2024; Munasinghe et al., 2024). Sa2VA connects a multimodal language model with SAM2 for unified image and video segmentation, while SPARROW augments video grounding models with target-specific tracked features and joint box and mask prompts (Yuan et al., 2025; Alansari et al., 2026). These methods ground a referred entity or generated phrase directly. VideoGLaMM derives grounded conversations and masks from existing video annotations, while Sa2VA constructs automatically labeled referring expressions for video objects (Yuan et al., 2025). VIDEOPLUM follows this broader direction but constructs temporally dense semantic-part pseudo-masks paired with their corresponding parent-object tracks.

Reference-guided video segmentation. Semi-supervised video object segmentation specifies a target using an annotated mask, typically in the first frame, and propagates it through the remaining frames. XMem uses complementary memory stores to preserve target information over long videos, while XMem++ retains one or more annotated frames and supports both complete objects and partial regions (Cheng & Schwing, 2022; Bekuzarov et al., 2023). SAM2 generalizes promptable segmentation to videos using clicks, boxes, or masks together with streaming memory (Ravi et al., 2024). SMITE addresses fine-grained video segmentation by learning the desired granularity from one or a few annotated reference images, which may be separate from the target video (Alimohammadi et al., 2024). PUMaVOS and SMITE consequently include challenging partial-region and semantic-part annotations. These methods can propagate object parts, but the target region is defined through visual annotations. In VIDEOPLUM, no annotated mask, box, or point is provided at inference, the requested semantic part must instead be inferred from the natural-language expression itself.

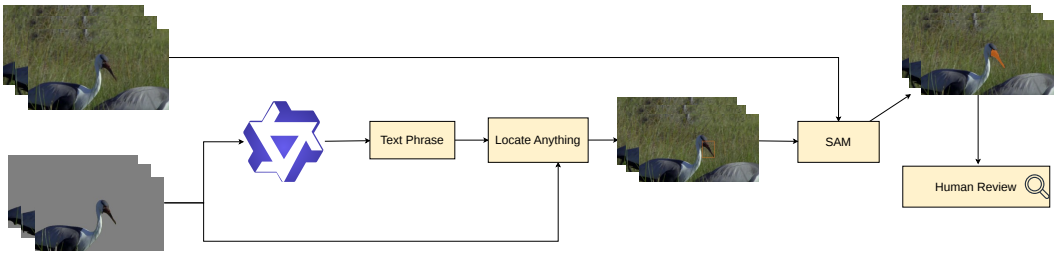


Figure 3: Data generation pipeline for part-level masks.

Our approach consists of two stages. We first construct temporally consistent semantic-part supervision from videos with object-level annotations, addressing the lack of dense part labels in existing datasets. We then use this supervision to train VIDEOPLUM, a language-guided video part grounding model built on a multimodal Qwen3.5-4B backbone and SAM-2

3 METHOD

3.1 STAGE 1: CONSTRUCTING TEMPORAL PART SUPERVISION

Dense semantic-part annotations are scarce in video datasets (Seo et al., 2020a; Khoreva et al., 2018; Yan et al., 2024; Bai et al., 2024), even when object-level tracks are available. We therefore construct temporally consistent part pseudo-labels from videos with a referring expression and an aligned parent-object track. Our corpus combines records derived from MeViS, Refer-YouTube-VOS, its GCG variant, and VidSTG (Seo et al., 2020a; Munasinghe et al., 2024; Zhang et al., 2020). we leverage Mevis' train split, Refer-YouTube-VOS-GCG train split, and all the splits of VidSTG. We then split these into 8,563 videos for training, 1,042 for validation, and 1,107 for internal testing. There is no match between the data in these 3 splits and the datasets we later to test on. The key idea is to constrain part discovery, localization, and tracking to the parent object, reducing background ambiguity and preventing associations across instances. Qwen lists the visible semantic parts, each with a canonical name, a visually specific phrase, and the number of instances that can appear at once (e.g., two hands), which bounds how many separate part tracks are formed.

We then localize and track each proposed part. For each parent-visible episode, we densely sample 16 frames for localization. LocateAnything (Wang et al., 2026) predicts part boxes on parent-isolated crops, and SAM-2 (Ravi et al., 2024) generates masks from the corresponding native-resolution boxes. We reject any detection whose mask lies mostly outside the parent object, since a part must belong to its object. We then link each part's detections into tracks. We compare pairs of detections that lie at most 16 frames apart in the video: SAM-2 propagates the earlier mask forward and the later mask backward, and the two detections are linked only if, in both directions, the propagated mask and the detected mask each cover more than half of the other. Chains of linked detections form a part track, and tracks supported by fewer than three detections are not used as pseudo-labels. Finally, SAM-2 fills in the frames between detections to obtain a dense part track. Each track records positive, absent, and unknown frames, with only positive and absent states contributing supervision. The resulting part tracks $(q_p, M^p_{1:T})_{p=1}^{P}$ contain part phrases, instance identities, and frame-level states. A sample of them undergoes human review before they are being used as training pseudo-labels; a manual quality assessment is provided in §4.

3.2 STAGE 2: LANGUAGE-GUIDED VIDEO PART GROUNDING

Given a video $V = \{I_t\}_{t=1}^{T}$, a parent-object expression o, and a requested-part expression p, VIDEOPLUM predicts the spatiotemporal mask of p within o.The model combines a multimodal Qwen3.5-4B backbone, fine-tuned with LoRA, with SAM-2. The multimodal Qwen3.5-4B backbone encodes the video frames and language request and generates a short response containing the requested-part phrase. A BIO span head identifies the answer tokens corresponding to this phrase. Let H_i^{ans} denote the final-layer hidden representation of answer token i. At inference time, the BIO head selects a requested-part span $\widehat{\mathcal{I}}_p$. We mean-pool the representations within this span and project them into the

Table 1: Comparison on SMITE, PUMaVOS, and EgoFun3D for video semantic-part segmentation. Avg. is the mean $\mathcal{J}\&\mathcal{F}$ over the three datasets. Best results are shown in **bold** and second-best results are underlined.

Method	SMITE			PUMaVOS			EgoFun3D			Avg.
	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$
Image Segmentation Baselines										
LISA-7B	16.00	20.97	18.49	22.09	24.04	23.06	15.14	23.49	19.32	20.29
LISA-13B	28.47	32.36	30.41	32.22	35.33	33.77	19.49	24.68	22.09	28.76
MLLM + SAM2 Pipelines										
PLUM + SAM2	32.12	37.36	34.74	46.06	48.29	47.17	27.37	31.64	29.50	37.14
GLaMM + SAM2	27.54	35.65	31.59	47.90	50.74	49.32	**33.09**	37.84	**35.46**	38.79
Qwen3.5-4B + SAM2	10.08	14.95	12.52	8.62	13.35	10.98	11.47	12.87	12.17	11.89
Video Segmentation Models										
LMPM	6.23	10.87	8.55	10.75	13.34	12.04	12.47	15.22	13.84	11.48
SAMWISE	6.33	16.84	11.59	20.61	22.68	21.64	14.81	19.92	17.37	16.87
VideoLISA	26.66	33.20	29.93	27.72	29.83	28.77	14.86	20.98	17.92	25.54
VideoLISA + XMem++	21.56	27.24	24.40	31.96	34.63	33.30	21.24	27.58	24.41	27.37
VideoGLaMM	10.79	16.46	13.63	22.46	26.45	24.45	19.31	23.74	21.52	19.87
Sa2VA-4B	20.51	24.48	22.49	40.29	45.80	43.05	27.24	33.32	30.28	31.94
Sa2VA-8B	28.57	32.96	30.77	40.67	45.26	42.97	28.16	35.46	31.81	35.18
VIDEOPLUM (Ours)	**58.55**	**65.26**	**61.91**	**54.14**	**58.97**	**56.55**	32.55	**38.26**	35.40	**51.29**

text-prompt input space of SAM-2:

$$z_p = \Pi\left(\frac{1}{|\widehat{\mathcal{I}}_p|}\sum_{i\in\widehat{\mathcal{I}}_p} H_i^{\text{ans}}\right), \qquad z_p \in \mathbb{R}^{256}, \tag{1}$$

where Π is a learned two-layer MLP. We augment the SAM-2 prompt encoder to accept z_p as a sparse prompt token in place of points or boxes. The same z_p is supplied at every frame, and SAM-2 combines it with the current frame's features and its memory of earlier predictions to decode a single mask per frame. Given the resized video frames X and the text-derived prompt z_p, the augmented SAM-2 model combines the prompt with frame-level visual features and streaming video memory to produce a mask-logit sequence:

$$\ell^p_{1:T} = \text{SAM2}^{\text{text}}_\phi(X, z_p), \qquad \ell^p_{1:T} \in \mathbb{R}^{T\times H\times W}, \tag{2}$$

where (H, W) denotes the spatial resolution of the predicted mask logits. SAM-2 uses its video memory to maintain the part track across frames.

Training. We train the model using language-generation, BIO span-identification, and video-mask objectives. Training uses teacher forcing: the decoder is conditioned on the reference response, and the annotated requested-part span provides supervision for the BIO head and is used to construct the mask prompt z_p. At inference time, the model generates the response and uses the BIO-predicted span instead.

For each training example, physical instances sharing the same semantic part phrase are merged to form the target semantic mask. Frames with known visible or absent targets contribute to the mask loss, while frames with unknown annotations are ignored. The overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \mathcal{L}^p_{\text{BIO}} + 2\mathcal{L}^p_{\text{BCE}} + 8\mathcal{L}^p_{\text{FT}}. \tag{3}$$

where $\mathcal{L}_{\text{LM}}$ and $\mathcal{L}^p_{\text{BIO}}$ are cross-entropy losses on the response tokens and BIO labels, and $\mathcal{L}^p_{\text{BCE}}$ and $\mathcal{L}^p_{\text{FT}}$ are binary cross-entropy and focal Tversky (Abraham & Khan, 2018) losses on the mask logits. At each step, the number of frames cycles through $\{4, 8, 12, 16\}$ (capped by the track length), and the frames are sampled uniformly over the track. We train with AdamW and cosine decay for 8 epochs on 16 GPUs (global batch 16), with learning rates of 2×10^{-5} for LoRA, 10^{-4} for the heads, and 5×10^{-5} for SAM-2. At inference, every method receives the same query, *Please segment the {part} of the object described as "{object}"* for parts and *Please segment the object described as "{expression}"* for objects; videos are processed in consecutive 16-frame windows.

Table 2: Comparison on Ref-DAVIS17 and MeViS Valid_u for video object segmentation. Best results are shown in **bold**, and second-best results are underlined.

Method	Ref-DAVIS17			MeViS Valid_u		
	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$
MLLM + SAM2 Pipelines						
GLaMM + SAM2	67.68	73.02	70.35	42.53	49.23	45.88
PLUM + SAM2	59.70	65.54	62.62	36.13	42.95	39.54
Video Segmentation Models						
VideoLISA	64.21	71.72	67.97	49.34	55.99	52.66
VideoGLaMM	59.41	68.97	64.19	43.93	52.49	48.21
VIDEOPLUM (Ours)	**71.84**	**78.41**	**75.12**	**53.57**	**61.26**	**57.42**

4 EXPERIMENTS

Implementation Details. We use Qwen3.5-4B (Qwen Team, 2026) as the multimodal backbone and initialize the segmentation decoder and temporal-memory modules from SAM2 Hiera-L (Ravi et al., 2024), its image encoder remains frozen. VIDEOPLUM is trained on MeViS (Ding et al., 2023), Refer-YouTube-VOS, its GCG-variant (Seo et al., 2020a; Munasinghe et al., 2024), and Vid-STG (Zhang et al., 2020), comprising 37,597 object and part requests. We sample 4, 8, 12, or 16 frames per example. Training uses a global batch size of 16 on 16 AMD MI210 GPUs.

Baselines. For video segmentation, we compare VIDEOPLUM with VideoGLaMM, VideoLISA, Sa2VA, SAMWISE, LMPM, and LISA (Munasinghe et al., 2024; Bai et al., 2024; Yuan et al., 2025; Cuttano et al., 2025; Ding et al., 2023; Lai et al., 2023). We also evaluate grounding-and-propagation baselines combining Locate Anything or Qwen3.5 with SAM 2, as well as PLUM and GLaMM masks propagated using SAM 2 (Blume et al., 2025; Rasheed et al., 2024). For image segmentation, we compare against LISA, GLaMM, PixelLM (Ren et al., 2023), VideoGLaMM, and Qwen3.5 with SAM 2.

Evaluation Datasets. For part-level video segmentation, we evaluate on SMITE (Alimohammadi et al., 2024), PUMaVOS (Bekuzarov et al., 2023), and EgoFun3D (Peng et al., 2026). For SMITE, we exclude nine non-text videos, which provide no language cues, and two unreleased videos, leaving 39 videos for evaluation. For video object segmentation, we use Ref-DAVIS17 (Khoreva et al., 2018) and a held-out local validation split of MeViS (Ding et al., 2023); no videos in this split were used for training or validation. For image part segmentation, we use PARTONOMY-Core (Blume et al., 2025). We evaluate video segmentation using region similarity $\mathcal{J}$ (mask IoU), contour accuracy $\mathcal{F}$ (boundary F-measure), and their mean $\mathcal{J}\&\mathcal{F}$.

Qualitative Evaluation. To evaluate the Stage 1 mask generation pipeline, shown in Figure 3, two human annotators assessed 100 videos sampled across three datasets: MeViS (33), VidSTG (34), and Ref-YouTube-VOS (33) (Ding et al., 2023; Zhang et al., 2020; Seo et al., 2020a). Each annotator independently evaluated a disjoint set of 50 videos, providing a binary judgment (reliable vs. unreliable) on the generated masks. Overall, the pipeline achieved a reliability rate of 92.0% (95% CI: 86.7%-97.3%), with no statistically significant difference between annotators (two-proportion z-test, $p = 0.461$), demonstrating that our Stage 1 predictions are sufficiently high quality to serve as training targets

4.1 VIDEO-PART SEGMENTATION

Language-guided semantic-part segmentation in video is not covered by a standard benchmark, so we adapt three datasets that provide complementary forms of part-level video annotation. SMITE (Alimohammadi et al., 2024) studies segmentation at arbitrary granularity and includes semantic decompositions of objects into regions such as body, wheel, and window and categories such as car, human, and horse. PUMaVOS (Bekuzarov et al., 2023) focuses on partial and unusual video masks, including object parts and regions whose boundaries may not coincide with strong visual cues. EgoFun3D (Peng et al., 2026) contains egocentric interaction videos with masks for functional object parts, represented as receptors and effectors such as a faucet handle and faucet spout.

Table 3: Micro- and macro-gIoU results on PARTONOMY-Core. Best results are shown in **bold**, and second-best results are underlined.

Method	Identification		Intersection		Difference		Part2Whole		Whole2Part	
	micro	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*	*micro*	*macro*
Image Segmentation Models										
SegLLM	**29.6**	**32.4**	32.2	33.8	**28.5**	**30.7**	**29.4**	**32.3**	**29.3**	**32.2**
LISA-13B	5.9	7.0	7.1	7.5	6.1	7.1	5.7	6.6	6.0	6.8
PixelLM-13B	6.8	8.4	4.4	4.8	4.2	4.8	4.6	5.4	6.3	7.8
GLaMM	5.3	5.9	5.9	6.2	5.2	6.0	4.8	5.6	4.9	5.8
PLUM-13B	14.5	27.4	23.7	29.9	14.9	24.8	14.3	26.8	15.4	27.5
Video Models										
VIDEOPLUM	28.3	30.2	**32.5**	**35.3**	25.1	26.7	28.0	29.6	27.4	29.1
VideoGLaMM	20.7	23.8	23.8	26.0	19.1	21.7	20.7	24.0	20.1	23.2

Together, these datasets cover conventional semantic parts, unusual partial regions, and interaction-defined functional parts. Results are provided in Table 1.

VIDEOPLUM achieves the best performance among the evaluated methods on SMITE and PUMaVOS, with $\mathcal{J}\&\mathcal{F}$ scores of 61.91 and 56.55, respectively. On SMITE, it outperforms the strongest baseline, PLUM + SAM2 (34.74), by 27.17 points. This substantial gain over an already part-aware pipeline highlights the advantage of performing part segmentation directly at the video level. On PUMaVOS, VIDEOPLUM surpasses GLaMM + SAM2 (49.32) by 7.23 points. On Ego-Fun3D, it nearly matches the best-performing method, reaching 35.40 compared with 35.46 for GLaMM + SAM2—a small numerical difference of only 0.06 points. Averaged across the three datasets, VIDEOPLUM achieves the highest score of 51.29, exceeding the strongest baseline average, GLaMM + SAM2 (38.79), by 12.50 points.

4.2 VIDEO OBJECT SEGMENTATION

We additionally evaluate VIDEOPLUM on standard referring video object segmentation to assess whether its part-oriented design preserves general object-level grounding. Table 2 reports results on Ref-DAVIS17 and MeVis' local validation split using $\mathcal{J}$, $\mathcal{F}$, and their average $\mathcal{J}\&\mathcal{F}$. Although VIDEOPLUM is designed primarily for fine-grained part segmentation, It performs strongly on standard referring object segmentation. On Ref-DAVIS17, it achieves a $\mathcal{J}\&\mathcal{F}$ score of 75.12, outperforming the strongest baseline, GLaMM + SAM2 (70.35), by 4.77 points and exceeding VideoLISA (67.97) and VideoGLaMM (64.19) by 7.15 and 10.93 points, respectively. On MeViS Valid_u, VIDEOPLUM achieves 57.42, surpassing the strongest baseline, VideoLISA (52.66), by 4.76 points and the strongest MLLM + SAM2 pipeline, GLaMM + SAM2 (45.88), by 11.54 points. These results demonstrate that the unified formulation supports segmentation at both part and object granularities without sacrificing object-level performance.

4.3 IMAGE PART-GROUNDING RESULTS

We evaluate image-level part understanding on PARTONOMY-Core (Blume et al., 2025), which tests compositional reasoning over objects and their parts. The benchmark comprises five tasks: Identification, Intersection, Difference, Part2Whole, and Whole2Part. It reports micro- and macro-gIoU. VIDEOPLUM achieves the best results on Intersection, with 32.5 micro-gIoU and 35.3 macro-gIoU, and ranks second on both metrics across the other four tasks. Compared with PLUM-13B, VIDEOPLUM improves on all five tasks, with gains of 8.8–13.8 points in micro-gIoU and 1.6–5.4 points in macro-gIoU. These results show that the unified representation retains strong static-image part grounding while extending part segmentation to video. Results are provided in Table 3

4.4 ABLATION STUDIES

Parent-object conditioning. We train a variant with an additional parent branch that predicts the parent-object span, segments the parent with SAM-2, and supplies its mask as a prompt to the part

Table 4: Ablations ($\mathcal{J}\&\mathcal{F}$). + *parent branch*: separately trained variant whose part decoding is conditioned on a predicted parent-object mask. − *temporal memory*: SAM-2 memory disabled at inference using the same weights. Mean is computed over the three part benchmarks.

	Parts				Objects
Variant	SMITE	PUMaVOS	EgoFun3D	Mean	Ref-DAVIS17
VIDEOPLUM	61.91	**56.55**	**35.40**	**51.29**	**75.12**
+ parent branch	**62.80** (+0.89)	55.46 (−1.09)	27.62 (−7.78)	48.63 (−2.66)	72.38 (−2.74)
− temporal memory	55.18 (−6.73)	51.56 (−4.99)	29.56 (−5.84)	45.43 (−5.86)	73.52 (−1.60)

Table 5: General video QA accuracy (%).

Model	**MVBench**	**Video-MME (short)**
Qwen3.5-4B (base)	62.03	68.00
VideoGLaMM	38.71	40.67
VIDEOPLUM	60.50 (−1.53)	63.33 (−4.67)

decoder; it follows the same recipe and checkpoint epoch. The branch slightly improves SMITE (+0.9 J&F) but hurts PUMaVOS (−1.1), Ref-DAVIS17 (−2.7), and especially EgoFun3D (−7.8), lowering the average over part benchmarks by 2.7 points. We attribute the EgoFun3D drop to unreliable parent masks in cluttered egocentric scenes, which then restrict the part prediction to the wrong region. VIDEOPLUM therefore grounds parts directly from their text span.

Temporal memory. Disabling SAM-2's memory at inference, so that each frame is decoded independently with the same weights, lowers performance on every benchmark: SMITE (−6.7 J&F), EgoFun3D (−5.8), PUMaVOS (−5.0), and Ref-DAVIS17 (−1.6). The smaller drop on Ref-DAVIS17 suggests that memory matters most for parts, which are smaller and more often occluded than the salient objects in that benchmark. Without memory, VIDEOPLUM still exceeds all LMM-based baselines on SMITE and PUMaVOS.

4.5 General Video Understanding

Fine-tuning for part segmentation could erode the backbone's general video reasoning. We therefore evaluate VIDEOPLUM on two multiple-choice video QA benchmarks, MVBench (Li et al., 2024) and the short split of Video-MME (Fu et al., 2024), and compare it with its base model, Qwen3.5-4B, and with VideoGLaMM (Munasinghe et al., 2024), another segmentation-oriented video LMM. All models receive the same 16 uniformly sampled frames and each benchmark's official prompt, and we take the option letter with the highest predicted probability. For MVBench we use the 19 tasks whose videos are publicly available (Fine-grained Pose requires the non-redistributed NTU RGB+D videos); Video-MME is evaluated without subtitles. As shown in Table 5, VIDEOPLUM retains most of the base model's accuracy (−1.5 on MVBench, −4.7 on Video-MME short) and outperforms VideoGLaMM by more than 20 points on both benchmarks.

4.6 Qualitative Results and Failure Analysis

Figure 4 shows three small parts from EgoFun3D (a lamp switch, a faucet handle and a faucet spout). VIDEOPLUM's masks closely match the ground truth (IoU 90–97), while VideoGLaMM and VideoLISA segment a region that misses the part (lamp switch) or only loosely cover it and spill onto the surroundings (faucet handle and spout). The gap holds over the full clips, where VIDEOPLUM's mean IoU is 80–90 versus at most 30 for either baseline. Similarly, Figure 2 shows that VIDEOPLUM accurately segments and tracks the microwave door, whereas the baselines fail to maintain the correct part mask.

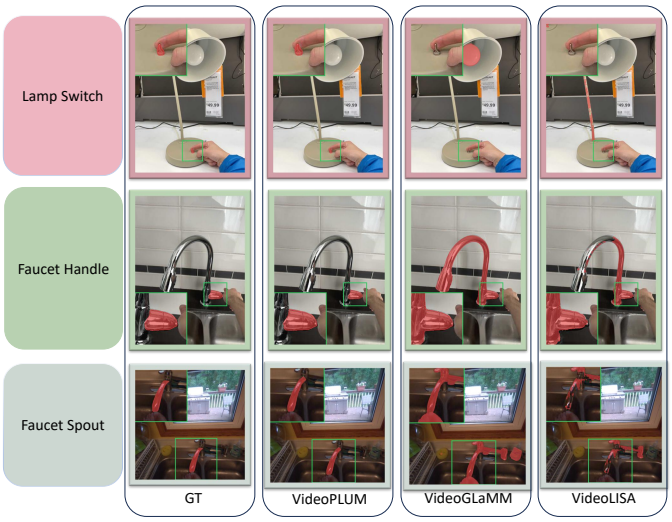


Figure 4: Qualitative part-segmentation results on EgoFun3D.

5 LIMITATIONS

While VIDEOPLUM substantially improves language-guided part grounding across diverse video benchmarks, several limitations remain. First, because dense part-level video annotations are scarce, our training data relies on automatically generated pseudo-labels produced by multiple pretrained models. Although our human evaluation finds these labels reliable in 92.0% of sampled videos (§4), errors in part discovery, localization, or temporal propagation can still introduce noise into training supervision. Second, part definitions and granularity can be inherently ambiguous: the same object may admit different semantic, structural, or functional decompositions depending on the query and context, while our training pipeline generates a fixed set of candidate parts for each object. Third, our evaluation covers semantic, unusual, and functional parts across three video benchmarks, but these datasets do not exhaust the diversity of objects, domains, temporal durations, and forms of part-level reasoning encountered in unconstrained videos.

6 CONCLUSION

In this work, we introduce VIDEOPLUM, a large multimodal model for language-guided, part-level visual grounding in videos. Our work addresses a key challenge in extending fine-grained visual grounding from images to video: a model must not only understand *which part* language refers to and localize it within the appropriate object, but also maintain that grounding as appearance, visibility, and viewpoint change over time. To make this possible without relying on dedicated segmentation tokens, VIDEOPLUM grounds ordinary generated language by identifying part-bearing spans and projecting their hidden representations directly into SAM2's prompt space. We further introduce an automatic data-generation pipeline that transforms existing object-level video annotations into temporally consistent semantic-part supervision, providing a practical route for training fine-grained video grounding models despite the scarcity of manually annotated part tracks. Across SMITE, PUMaVOS, and EgoFun3D, VIDEOPLUM substantially improves over existing image- and video-grounding approaches, while retaining strong performance on standard video object segmentation and transferring effectively to image-level compositional part grounding. Together, these results demonstrate that fine-grained language understanding and temporally persistent visual grounding can be learned within a unified multimodal model, moving video understanding beyond tracking *which object* is being discussed toward grounding *which part of that object* language refers to over time. We hope this provides a foundation for future models that reason about increasingly fine-grained, compositional, and functional elements of dynamic visual environments.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets (specifically, using Qwen3.5-4B and SAM-2 to construct the part pseudo-labels for training). We have not used generative AI for any other required-disclosure tasks, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI to edit the research paper to improve readability. All AI-assisted work was rigorously reviewed. Generated pseudo-labels underwent automatic filtering and sampled manual assessment, and all AI-assisted text was verified by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation, 2018. URL https://arxiv.org/abs/1810.07842.

Mohamad Alansari, Naufal Suryanto, Divya Velayudhan, Sajid Javed, Naoufel Werghi, and Muzammal Naseer. Sparrow: Learning spatial precision and temporal referential consistency in pixel-grounded video mllms, 2026. URL https://arxiv.org/abs/2603.12382.

Amirhossein Alimohammadi, Sauradip Nag, Saeid Asgari Taghanaki, Andrea Tagliasacchi, Ghassan Hamarneh, and Ali Mahdavi Amiri. Smite: Segment me in time, 2024. URL https://arxiv.org/abs/2410.18538.

Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Lei Liu, Zheng Zhang, and Mike Zheng Shou. One token to seg them all: Language instructed reasoning segmentation in videos. *arXiv preprint arXiv:2409.19603*, 2024.

Maksym Bekuzarov, Ariana Bermudez, Joon-Young Lee, and Hao Li. Xmem++: Production-level video segmentation from few annotated frames, 2023.

Ansel Blume, Jeonghwan Kim, Hyeonjeong Ha, Elen Chatikyan, Xiaomeng Jin, Khanh Duy Nguyen, Nanyun Peng, Kai-Wei Chang, Derek Hoiem, and Heng Ji. PARTONOMY: Large multimodal models with part-level visual understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=yjLew3Nd7z.

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 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pp. 1979–1986, USA, 2014. IEEE Computer Society. ISBN 9781479951185. doi: 10.1109/CVPR.2014.254. URL https://doi.org/10.1109/CVPR.2014.254.

Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022.

Claudia Cuttano, Gabriele Trivigno, Gabriele Rosi, Carlo Masone, and Giuseppe Averta. Samwise: Infusing wisdom in sam2 for text-driven video segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 3395–3405, June 2025.

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

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. doi: 10.1109/TPAMI.2009.167.

Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees G. M. Snoek. Actor and action video segmentation from a sentence, 2018. URL https://arxiv.org/abs/1803.07485.

Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. *arXiv preprint arXiv:2112.00933*, 2021.

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, 2021. URL https://arxiv.org/abs/2106.09685.

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*, pp. 123–141, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-20869-1. doi: 10.1007/978-3-030-20870-7_8. URL https://doi.org/10.1007/978-3-030-20870-7_8.

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.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark, 2024. URL https://arxiv.org/abs/2311.17005.

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 (ICCV)*, pp. 920–930, October 2023.

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 2311.13435*, 2023.

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

Weikun Peng, Denys Iliash, and Manolis Savva. EgoFun3D: Modeling Interactive Objects from Egocentric Videos using Function Templates. *arXiv preprint arXiv:2604.11038*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, Amir Mousavi, Yiwen Song, Abhimanyu Dubey, and Dhruv Mahajan. PACO: Parts and attributes of common objects. In *arXiv preprint arXiv:2301.01795*, 2023.

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

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

Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model, 2023.

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*, pp. 208–223, Berlin, Heidelberg, 2020a. Springer-Verlag. ISBN 978-3-030-58554-9. doi: 10.1007/978-3-030-58555-6_13. URL https://doi.org/10.1007/978-3-030-58555-6_13.

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*, pp. 208–223, Berlin, Heidelberg, 2020b. Springer-Verlag. ISBN 978-3-030-58554-9. doi: 10.1007/978-3-030-58555-6_13. URL https://doi.org/10.1007/978-3-030-58555-6_13.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, Jan Kautz, Lei Zhang, and Zhiding Yu. Locateanything: Fast and high-quality vision-language grounding with parallel box decoding. *arXiv preprint arXiv:2605.27365*, 2026.

XuDong Wang, Shaolun Zhang, Shufan Li, Konstantinos Kallidromitis, Kehan Li, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Segllm: Multi-round reasoning segmentation. *arXiv preprint arXiv:2410.18923*, 2024.

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*, pp. 2761–2770, 2023.

Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation, 2022. URL https://arxiv.org/abs/2201.00487.

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

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

Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences, 2020. URL https://arxiv.org/abs/2001.06891.

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*, 2017.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

Jiawen Zhu, Zhi-Qi Cheng, Jun-Yan He, Chenyang Li, Bin Luo, Huchuan Lu, Yifeng Geng, and Xuansong Xie. Tracking with human-intent reasoning, 2023.

A Appendix

A.1 Stage 1: Part Pseudo-Label Construction

Sources. The corpus is built from videos that come with a referring expression and an aligned object track: the training split of MeViS, the training split of the GCG variant of Refer-YouTube-VOS, and VidSTG (all annotation splits), whose object boxes we convert to masks with SAM-2. No validation or test videos of MeViS or Refer-YouTube-VOS are used.

Part proposal. For each object track, Qwen3.5-4B (Team, 2026) lists the visible semantic parts of the object. Each proposal contains a canonical part name, a visually specific part phrase, and the maximum number of instances of that part visible at the same time (e.g., two hands). The proposal prompt asks only for visible regions that are smaller than, and belong to, the object.

Localization. For each episode in which the object is visible, 16 frames are sampled for localization. LocateAnything (Wang et al., 2026) predicts part boxes on crops around the object, and SAM-2 converts each box into a mask at the original resolution. Detections with at most half of their mask pixels inside the object mask are rejected.

Temporal association. Detections of the same part are compared in pairs at most 16 frames apart. SAM-2 propagates the earlier mask forward and the later mask backward, and two detections are linked only if, in both directions, the propagated mask and the detected mask each cover more than half of the other. Linked detections form part tracks, with at most the proposed instance count of tracks per part; when a part may have several instances, or several detections occur in the same frame, links must additionally be transitively consistent. Tracks supported by fewer than three detections are discarded. SAM-2 then propagates masks between the detected frames to obtain dense tracks. Each frame of a track is labeled visible, absent, or unknown, and unknown frames are excluded from the mask loss.

A.2 Corpus Statistics

The corpus is split into training, validation, and internal test partitions with no video shared between partitions (Table 6). Each track yields one whole-object request and one request per semantic part; instances sharing a part phrase are merged into a single target.

Table 6: Stage-2 corpus. Requests = whole-object + part requests.

Partition	Source	Videos	Object req.	Part req.	Total req.
Train	MeViS	742	742	2,042	2,784
	Ref-YTVOS (GCG)	3,830	3,830	12,475	16,305
	VidSTG	3,991	3,991	14,517	18,508
	Total	8,563	8,563	29,034	37,597
Validation	MeViS	89	89	236	325
	Ref-YTVOS (GCG)	497	497	1,731	2,228
	VidSTG	456	456	1,609	2,065
	Total	1,042	1,042	3,576	4,618
Test	*Total*	1,107	1,107	3,770	4,877

A.3 Stage 2: Model and Training

Architecture. The backbone is Qwen3.5-4B, fine-tuned with LoRA adapters on the attention and linear-attention projections of the language model and on the vision merger; all other backbone weights are frozen. The BIO head is a two-layer MLP ($2560 \to 2560 \to 3$) over answer-token hidden states, and the span-to-prompt projection Π is a two-layer MLP ($2560 \to 2560 \to 256$, ReLU). The segmenter is SAM-2 (Hiera-L); its image encoder is frozen, and its mask decoder, memory attention, memory encoder, and no-mask embedding are trained. The prompt z_p is given to SAM-2 at every frame, and SAM-2 predicts a single mask per frame using its memory of earlier frames.

Optimization. Hyperparameters are listed in Table 7. Training uses teacher forcing: the reference response conditions the decoder and the annotated span constructs z_p. The number of frames per step cycles through $\{4, 8, 12, 16\}$ (capped by the track length), with frames spaced uniformly over the track, and SAM-2 tracks half of the training samples in reverse temporal order. The focal Tversky loss is computed per frame as $(1 - \mathrm{TI})^{\gamma}$ with $\mathrm{TI} = \mathrm{TP}/(\mathrm{TP} + \alpha\,\mathrm{FN} + \beta\,\mathrm{FP})$.

Threshold selection. The mask-logit threshold is selected on the validation partition from $\{-2.0, -1.75, -1.5, -1.25, -1.0, -0.75, -0.5, 0\}$, ranking by gIoU averaged over sources with part and object requests weighted equally; the selected value, -1.5, is used for all variants. No evaluation benchmark is used for any selection.

Table 7: Stage-2 hyperparameters.

LoRA	r=16, α=32, dropout 0.05
Optimizer	AdamW, weight decay 0
Learning rates	LoRA 2×10^{-5}; heads 10^{-4}; SAM-2 5×10^{-5}
Training	8 epochs (18,800 steps)
Schedule	17 warm-up steps, then cosine decay defined over 23,500 steps
Batch	16 GPUs, one request per GPU per step
Gradient clipping	1.0
Precision	bf16 backbone
Frames per step	cycling $\{4, 8, 12, 16\}$, uniform over the track
Qwen video resolution	4,096 to 1,048,576 pixels per frame
Loss weights	$\mathcal{L}_{LM}$: 1, $\mathcal{L}_{BIO}$: 1, $\mathcal{L}_{BCE}$: 2, $\mathcal{L}_{FT}$: 8
Focal Tversky	α=0.3 (FN), β=0.7 (FP), γ=0.75
Seed	42

A.4 Inference

LMM-based methods receive the query *Please segment the* {*part*} *of the object described as "*{*object*}*".* for parts and *Please segment the object described as "*{*expression*}*".* for objects. VideoPLUM decodes greedily with at most 48 new tokens, and the BIO head selects the part span in the generated response. Videos are processed in consecutive 16-frame windows covering every frame, and masks are obtained by thresholding the SAM-2 logits at -1.5.

B Evaluation Protocol

Benchmarks. Predictions are made on every frame of each video. **SMITE**: the paper describes 50 videos (41 semantic and 9 Non-Text), but the public archive contains 48; two semantic videos are absent and cannot be identified. We exclude the Non-Text videos, whose regions are defined by reference masks rather than part names. **PUMaVOS**: all 24 videos, scored at annotation resolution. **EgoFun3D**: all 271 videos listed in the release, each part role scored as an independent binary mask. **Ref-DAVIS17** and **MeViS** (valid_u, 50 videos, 793 expressions) for whole objects. We report $\mathcal{J}$, $\mathcal{F}$, and $\mathcal{J}\&\mathcal{F}$.

Multiple parts per video. For SMITE and PUMaVOS, where each pixel has one label, each method's per-query masks are combined by assigning every pixel to the query with the largest positive margin over that method's threshold (background otherwise; ties to the lower class id). The same rule is applied to all methods.

Train–test overlap audit. To verify that no benchmark video appears in our training data, we compare frame content rather than file hashes, so that re-encoded or rescaled copies are also detected. All 129,596 frames referenced by the training and validation partitions, and evaluation frames (all frames of Ref-DAVIS17; 16 evenly spaced frames per video for SMITE, PUMaVOS, EgoFun3D, and MeViS), are reduced to 32×32 grayscale thumbnails. Each evaluation frame is matched to its nearest training frame by cosine similarity, and a match is confirmed if the mean absolute pixel difference is below $3/255$. No evaluation frame of any reported benchmark matches a training frame. The audit detects whole-frame copies; it does not detect crops or flips.

C General Video Understanding

All models receive the same 16 frames and each benchmark's official prompt, and the prediction is the answer letter with the highest next-token probability after the prompt's answer prefix. For MVBench, frames are sampled at segment midpoints within each question's time span; the Fine-grained Pose task is excluded because its NTU RGB+D videos are not redistributed, leaving 19 tasks (3,800 questions). For Video-MME we use the short split (300 videos under 2 minutes, 900 questions) without subtitles, with frames sampled uniformly.