TrajTok: Learning Trajectory Tokens Enhances Video Understanding

Supplementary Material

1. Segementer Training Details

In the **TrajAdapter** and **TrajVLM** settings, we pretrain the trajectory segmenter once and reuse its weights for initialization during downstream probing and VLM training. This section provides full details of the dataset construction, annotation pipeline, filtering criteria, and training configuration for our segmenter training..

1.1. Dataset Construction

Sources. We construct a video & image corpus for segmenter training by combining: Panda (video) [3], CC12M (image) [1], CC3M (image) [10], and a subset of DataComp-50M (image) [6]. All samples are annotated with pseudo panoptic trajectory masks using the TrajViT trajectory–generation pipeline, followed with data filtering. We describe the details below.

Annotation Pipeline. We adopt the same annotation process as in the TrajViT paper [12]. In summary, the pipeline consists of four steps.

1. sample frames and detect keyframes based on feature changes in colorspace and Luminance Histogram.
2. generate panoptic object masks in the key frames using DirectSAM [2] model.
3. track objects across frames via SAM2 [9].
4. merge instance masks between CLIPs using heuristics like IOU overlaps to form long-term trajectories.

The pipeline uses external models like DirectSAM and SAM2 [2, 9]. For images, only spatial segmentation steps are applied.

Quality Filtering. We apply two filtering criteria to remove low-quality pseudo labels:

- **Coverage filter:** remove samples where the union of all trajectory masks covers less than 80% of pixels.
- **Object-count filter:** remove samples containing fewer than 10 detected objects.

After filtering, we retain roughly 2.5M images and 2.0M videos for segmenter pretraining.

1.2. Training Configuration

The segmenter is trained on the filtered dataset. Different from TrajViT2 where all modules are trained from scratch, we initialize the ConvNext-small patch encoder from DINOv3's weights, which helps in the generalization performance of produced segments. Other modules are initialized from scratch. We train the model for 20 epochs with 8 A100 GPUs. We use the base learning rate of 1×10^{-3}, and adopt a linear decay learning-rate schedule with warm-up. Additional hyperparameters are summarized in Table 1.

Hyperparameter	Value
Resolution	224
Frame sampling	uniform 16 frames (videos only)
Optimizer	AdamW
Base LR	1×10^{-3}
Weight decay	0.01
Optimizer momentum	$\beta_1 = 0.9,\ \beta_2 = 0.999$
Batch size	video-128, image-512
Training epochs	20
LR schedule	linear decay
Warm up epochs	1
Warm up schedule	linear warm-up
Random crop scale	(0.2, 1.0)
Random crop ratio	(3/4, 4/3)
Horizontal flip probability	0.5
Color jitter probability	0.8
Gaussian blur probability	0.5
Grayscale probability	0.2

Table 1. **Hyperparameters used for segmenter pretraining.**

2. More Qualitative Examples of the Segmenter

We show examples of generated trajectories from our training set in Fig. 1. Overall, the segmenter exhibits strong semantic grouping ability, consistently discovering object-level regions that are sufficiently accurate for downstream understanding tasks. From the perspective of pixel-level segmentation quality, however, the lightweight design and low output resolution introduce several expected limitations: the model occasionally misses very small objects, may over-merge background regions, and produces imprecise object boundaries. These imperfections, while noticeable visually, do not hinder its effectiveness as a trajectory proposal module, as our downstream tasks primarily rely on correct semantic grouping rather than pixel-perfect masks.

3. Quantitative Evaluation of the Segmenter

Although the proposed segmenter in the main paper is intentionally lightweight—prioritizing semantic grouping over pixel-level precision—we additionally study how well it can perform on the standard panoptic video segmentation task when its capacity is scaled up. This experiment is conducted purely for analysis and is *not* used by any model in the main paper.

Scaling up the segmenter. We keep the same training dataset as described in Sec. 1.1, but increase the segmenter capacity in two ways: (1) replacing the ConvNeXt-Tiny patch encoder with a ConvNeXt-Large backbone and ex-

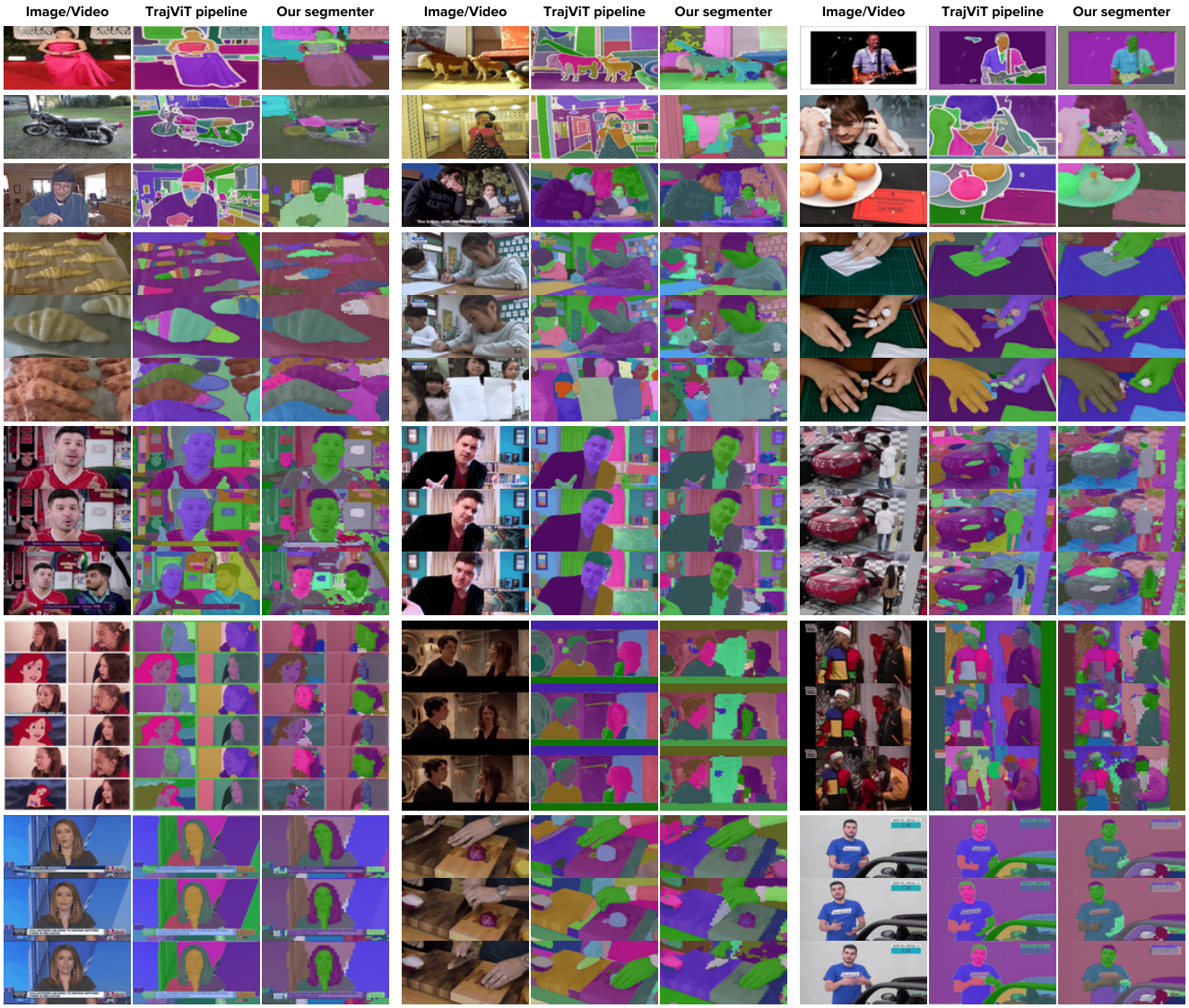

Figure 1. Qualitative Examples of the trajectory masks produced by our segmenter.

panding the Perceiver stack from 2 layers to 4 layers, and (2) producing full-resolution predictions by adding a pixel decoder identical to the one used in SAM [9], applied on top of the downsampled patch features. The input and output resolution are both set to 512×512.

Benchmark and competitors. We evaluate on the ViPEntitySeg [8] benchmark and compare against two state-of-the-art video panoptic segmentation systems: EntitySAM [11] and SAM 2.1 [9]. We report VEQ-SQ, VEQ-RQ, and STQ-EN following the benchmark protocol.

Results. Table 2 shows that the scaled-up version of our segmenter achieves competitive performance, surpassing EntitySAM in VEQ-SQ and improving VEQ-RQ relative to SAM 2.1. While its STQ-EN score is slightly lower than EntitySAM, these results demonstrate that our grouping-centric design can approach state-of-the-art performance when augmented with a strong visual backbone and a full-resolution decoder, confirming our segmenter design is reasonable.

VIPSeg Benchmark	VEQ-SQ	VEQ-RQ	STQ-EN
EntitySAM	84.7	**64.5**	**43.3**
SAM 2.1	83.1	36.7	41.7
Ours (scaled-up)	**85.5**	45.1	40.2

Table 2. **Evaluation of a scaled-up version of our segmenter on the ViPEntitySeg benchmark.** This model is *not* used in any main-paper experiments; it serves only as an isolated study of segmentation quality under increased capacity.

4. Training Details for TrajViT2

For all TrajViT2 experiments and baseline models, we optimize using the AdamW optimizer [7] with a base learning rate of 10^{-4}, weight decay of 10^{-2}, and mixed-precision training. We use a cosine annealing schedule with a linear warm-up of one epoch. The contrastive batch size is 128 for video clips and 1024 for images. All models are trained for 20 epochs using 8 NVIDIA A100 GPUs. During training, we apply standard video augmentation including random ColorJitter, Grayscale, Gaussian blur, horizontal flip, and resized cropping. At evaluation, we use only a single resizing operation for consistency. All models adopt a ViT-Large transformer and operate on 224-resolution inputs with 16 uniformly sampled frames.

5. Training Details for TrajAdapter

For all TrajAdapter experiments, we follow the standard protocol for probing pretrained video encoders. We use the AdamW optimizer with a learning rate of 1×10^{-4} and weight decay of 0.5. The pretrained backbone is kept frozen, while the trajectory encoder and probing head are updated. Before classification, video features are layer-normalized. We train with a batch size of 128 for 10 epochs. This configuration is used for all TrajAdapter experiments on both Kinetics-400 and Something-Something-V2 probing tasks.

6. Training Details for TrajVLM

We provide more training details for TrajVLM in this section.

Data sources. As discussed in the main paper, TrajVLM is trained using a two-stage procedure. For the pretraining stage, we closely follow the Molmo training paradigm [5] and use the same PixMo captioning split to align visual representations with the language model. For the instruction-tuning stage, we adopt the mixture of public academic VideoQA datasets and synthetic QA pairs curated in Molmo-2 [4]. These datasets span a wide range of reasoning skills—including temporal grounding, causal inference, long-horizon understanding, and multi-step procedural reasoning—and are summarized in Table 3. In total, the mixture contains approximately 5 million training examples.

Training hyperparameters. The training hyperparaemters of the first stage follows Molmo [5]. We summarize the training hyperparameters for the second stage at Table 4.

Table 3. **Datasets used for training TrajVLM under the "final" mixture.** This mixture combines a large set of academic VideoQA datasets, temporal reasoning datasets, and synthetic captioning/QA corpora.

Category	Dataset Name(s)	Notes / Source
Academic VideoQA	`llava_video_mc_academic`	MC-style QA
	`llava_video_oe_academic`	Open-ended QA
	`clevrer`	Causal & counterfactual reasoning
	`funqa`	Fine-grained temporal QA
	`star`	Long-horizon procedural QA
	`intent_qa`	Human intent reasoning
	`tgif`	Action/state transition QA
	`video_localized_narratives`	Localized narrations
	`road_text_vqa`	Driving VQA
	`countix_oe`	Counting QA (open-ended)
	`camerabench_qa`	Camera-motion VQA
Action / Activity QA	`nextqa_mc`	Next-QA multiple-choice
	`news_video_qa_filtered`	News comprehension QA
	`how2qa`	How-to instructional QA
	`sutd_trafficqa`	Traffic event QA
	`social_iq2`	Social reasoning
	`sportsqa_oe`	Sports QA (OE)
	`cinepile`	Movie understanding QA
	`ssv2_qa`	Something-Something QA
	`moments_in_time_qa`	Activity recognition QA
	`kinetics_qa`	Kinetics QA
	`charades_sta_all_qa`	Charades Spatial-Temporal QA
	`coin_all_qa`	Procedural task step QA
Video Captioning / Highlighting	`youcook2_all_qa`	Recipe video QA/caption
	`activitynet_all_qa`	ActivityNet QA/caption
	`ego4d_all`	Ego4D narrations + QA
	`video_localized_narratives_caption`	Captioning corpus
	`qv_highlights`	Highlight detection w/ text
	`motionbench_train`	Long-range motion reasoning
Internal / Synthetic VideoQA	`vixmo_syn_video_capqa_v2`	200K synthetic QA pairs
	`vixmo3_top_level_captions_min_3`	101K curated human captions
	`vixmo_clip_qa_all`	CLIP-constructed QA corpus

Hyperparameter	Value
Max video frames	128
Total training steps	10,000
Training stages	Pretraining + Instruction tuning
Sequence length	8192
Global batch size	32
Device batch size	4
Number of GPUs	$8 \times$ A100 (80GB)
Precision	bfloat16 (AMP)
Optimizer	AdamW
LLM learning rate	1×10^{-5}
ViT learning rate	5×10^{-6}
Connector learning rate	5×10^{-6}
Learning rate warmup	200 steps
LR schedule	Multimodal cosine decay
Weight decay	0.0 (LLM / ViT / connector)
Adam momentum	$\beta_1 = 0.9,\ \beta_2 = 0.95$
Adam ϵ	1×10^{-6}
Gradient clipping	1.0

Table 4. **Key hyperparameters used for training TrajVLM.** Values reflect the shared configuration across all VLM experiments.

References

[1] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021. 1

[2] Delong Chen, Samuel Cahyawijaya, Jianfeng Liu, Baoyuan Wang, and Pascale Fung. Subobject-level image tokenization. *arXiv preprint arXiv:2402.14327*, 2024. 1

[3] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple

cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024. 1

[4] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Rohun Tripathi, Sangho Lee, Mohammadreza Salehi, Jason Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Ali Farhadi, and Ranjay Krishna. Molmo 2: Open weights and open data for state-of-the-art video and image models, 2025. Technical Report. 3

[5] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. *arXiv preprint arXiv:2409.17146*, 2024. 3

[6] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023. 1

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[8] 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. 2

[9] Nikhila Ravi et al. Sam 2: Segment anything in images and videos. *arXiv:2408.00714*, 2024. 1, 2

[10] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 1

[11] Mingqiao Ye, Seoung Wug Oh, Lei Ke, and Joon-Young Lee. Entitysam: Segment everything in video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24234–24243, 2025. 2

[12] Chenhao Zheng, Jieyu Zhang, Mohammadreza Salehi, Ziqi Gao, Vishnu Iyengar, Norimasa Kobori, Quan Kong, and Ranjay Krishna. One trajectory, one token: Grounded video tokenization via panoptic sub-object trajectory. *arXiv:2505.23617*, 2025. 1