Learning Streaming Video Representation via Multitask Training

Supplementary Material

1. Pre-training Datasets

We list StreamFormer's pre-training datasets in Table 7. Note that for our Video Object Segmentation datasets, we utilize the VIS datasets and VIS-style produced COCO-pseudo videos for their relatively well correspondence within the objects inside the videos or pseudo videos.

Task	Pre-training Dataset	Scale
Global-level		
AR	K400 [12], SSv2 [38]	400K
VTR	MSRVTT [107], MSVD [14], ActivityNet [11], DiDeMo [2], LSMDC [88], VATEX [101]	94K
Temporal-level		
TAL	ActivityNet-1.3 [11], FineAction [67], HACS [126]	180K
TVG	CharadesSTA [37], QVHighlights [52], TaCoS [82], ANet-Captions [51], DiDeMo [2], QuerYD [72]	120K
Spatial-level		
VOS	YouTubeVIS-19 [113], LVVIS [95], COCO [60]	120K
RVOS	MEVIS [26], Refer-YouTube-VOS [84]	36K
Total	-	∼1M

Table 7. **Pretraining datasets.** The abbreviations are defined as follows: "AR" for action recognition, "VTR" for video-text retrieval, "TAL" for temporal action localization, "TVG" for temporal video grounding, "VOS" for video object segmentation, and "RVOS" for referring video object segmentation.

2. Downstream Implementation Details

2.1. Online Action Detection

In Online Action Detection, we adopt MAT [96] as our baseline detection head due to its strong detection performance. Following prior works [42, 110, 129], we extract the video frames at 24 fps, using a video chunk size of 6 (*i.e.*, 4 features per second). To ensure a fair comparison, we also incorporate optical flow features extracted using the BN-Inception model [47], which are concatenated with visual features and fed into the OAD model. We train the model using Adam optimizer for 25 epochs, with a batch size of 16 and a learning rate of 7×10^{-5}. All experiments are conducted on a single NVIDIA A100 GPU.

2.2. Online Video Instance Segmentation

For Online Video Instance Segmentation, we use CTVIS [116] as our baseline, and keep all head modules' hyper parameters the same. For our backbone, we train and test all frames resizing the short side to 224p. The batch size per device is 8 with an initial learning rate of 0.0001 and decays at 12,000 and 24,000 iterations, respectively on 4 NVIDIA H100 GPUs.

2.3. Video Question Answering

For Video Question Answering task, we adopt the LLaVA-NeXT-Video [124] framework as our code base. To make our comparisons fair, we use the same model architecture as LLaVA-NeXT (Vicuna-1.5-7B [131]) except for our StreamFormer (SigLIP-ViT-Base-Patch16-224 [119]) replacing the original CLIP-Vit-Large-Patch14-336 [78]. Note that our StreamFormer is initialized from SigLIP and multi-tasking trained with SigLIP's Text Encoder, thus making it still suitable for pre-training and supervised fine-tuning on image-text data. For image datasets, we use the LAION/CC/SBU 558K [64]

for pretraining, LLaVA-NeXT-Data 779K [124] for supervised fine-tuning on image-text. For Video dataset, we sample 10% from LLaVA-Video-178K's [125] academic and YouTube short videos (0 - 60s), resulting in a total 86.4K video-text samples. For model evaluation, we use the lmms-eval toolkit [122] with 16 frames per video. We run all experiments of VideoQA on 4 NVIDIA H100 GPUs. The detailed configuration is shown in Table 8.

		Pre-Training	Instruction Tuning	
			Image-Text	Video
Vision	**Resolution**	224	224×{2×2, 1×{2,3}, {2,3}×1}	224×{2×2, 1×{2,3}, {2,3}×1}
	#Tokens	196	Max 196×5	Max 196×5
Data	**Dataset**	LCS [64]	LLaVA-Next 779K [65]	LLaVA-Video-178K subset [125]
	#Samples	558K	779K	86K
Model	**Trainable**	Projector	Projector + LLM	Projector + LLM
	7B Vicuna 1.5 LLM	20.0M	6.8B	6.8B
Training	**Batch Size**	256	128	128
	LR: ψ_{vision}	N/A	N/A	N/A
	LR: $\{\theta_{\text{proj}}, \phi_{\text{LLM}}\}$	1×10^{-3}	2×10^{-5}	2×10^{-5}
	Epoch	1	1	1

Table 8. Detailed configuration for each training stage of our LLaVA-NeXT (StreamFormer).

3. More Results

We add more results of StreamFormer in this section.

3.1. Video Action Recognition

In Table 9, we add more comparisons with recent causal and autoregressive models, including TRecViT [32] and Toto [81] on Video Action Recognition tasks of K400 and SSv2. While StreamFormer achieves stronger results on K400 and is competitive on SSv2, we emphasize that StreamFormer's key advantage is **multitask learning across different spatio-temporal granularities**—from short-term action recognition to fine-grained, frame-level tasks like OAD and Online VIS.

3.2. Downstream Evaluations

In Table 10, we continue to supplement more results with TVSeries [24] (w/o flow) for Online Action Detection and OVIS [76] (224p and w/o COCO pre-training) for Online Video Instance Segmentation.In Table 11, we detail our results in Real-Time Visual Understanding task of StreamingBench [31]. Note that our LLM is Vicuna-7B-v1.5.

3.3. Qualitative Results

We also add qualitative results of StreamFormer's Online Video Instance Segmentation in Figure 4. Frames are sampled from the validation set of Youtube-VIS 2019.

Backbone	K400	SSv2
Toto	64.7	-
TRecViT	78.4	66.8
StreamFormer	82.2	66.5

Table 9. Comparison to other causal backbones.

Backbone	TVSeries	OVIS
	mAP	AP
SigLIP	84.8	18.2
StreamFormer	88.1	20.8

Table 10. Comparison on additional downstream datasets.

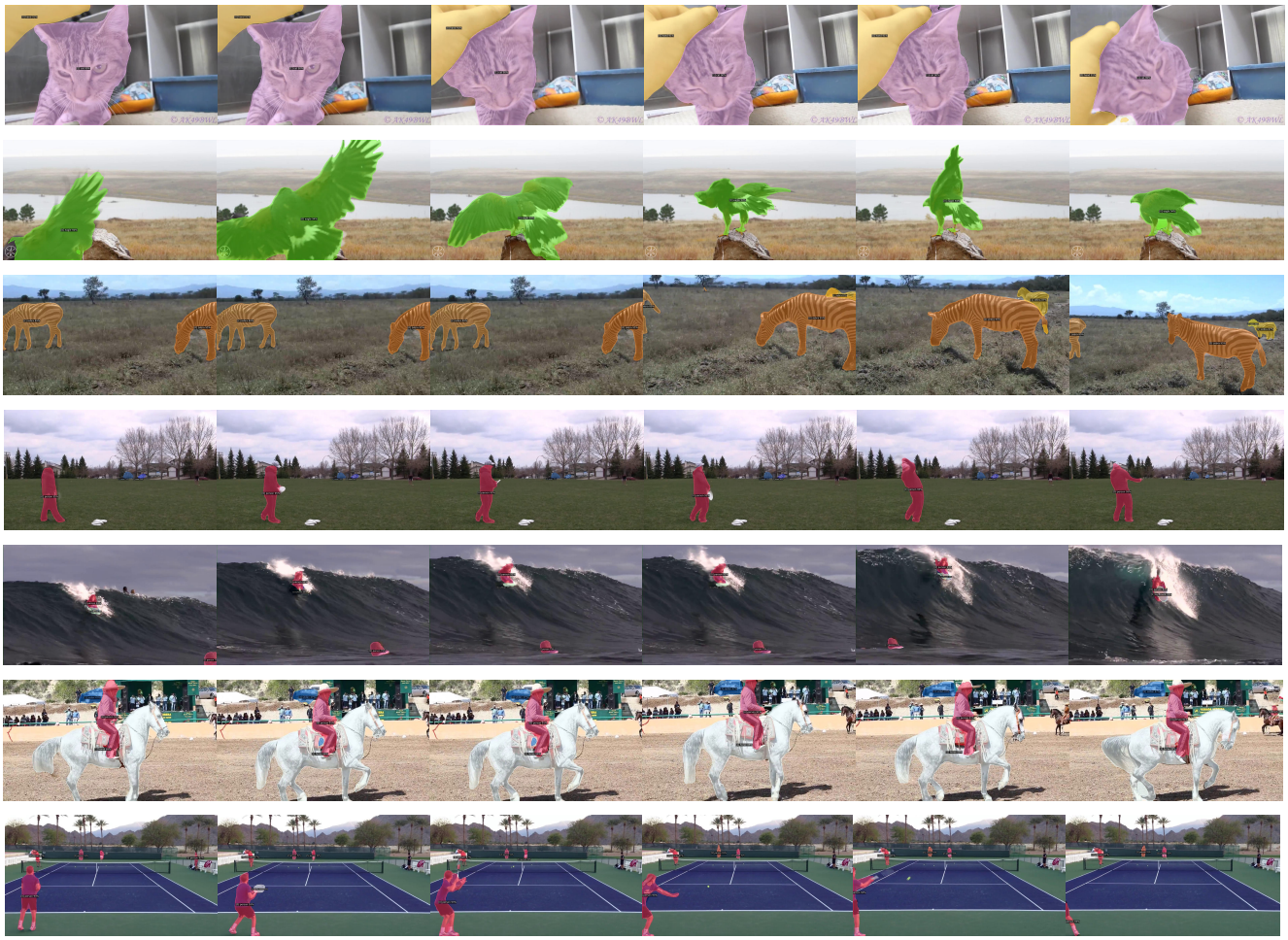

Figure 4. Qualitative results of Online Video Instance Segmentation. Zoom in for a better view.

Model	Frames	StreamingBench Real-Time Visual Understanding										
		OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	**All**
InternVL-V2	16	68.12	60.94	69.40	77.12	67.70	62.93	59.26	53.25	54.96	56.48	63.72
Kangaroo	64	71.12	84.38	70.66	73.20	67.08	61.68	56.48	55.69	62.04	38.86	64.60
LongVA	128	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96
VILA-1.5	14	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32
Video-CCAM	96	56.40	57.81	65.30	62.75	64.60	51.40	42.59	47.97	49.58	31.61	53.96
Video-LLaMA2	32	55.86	55.47	57.41	58.17	52.80	43.61	39.81	42.68	45.61	35.23	49.52
LLaVA-Next (**StreamFormer**)	16	69.16	66.67	63.10	57.40	70.2	50.00	67.86	52.8	66.67	12.5	58.24

Table 11. Performance comparison on StreamingBench Real-Time Visual Understanding tasks. The abbreviations are defined as follow: "OP" for Object Perception, "CR" for Causal Reasoning, "CS" for Clips Summarization, "ATP" for Attribute Perception, "EU" for Event Understanding, "TR" for Text-Rich Understanding, "PR" for Prospective Reasoning, "SU" for Spatial Understanding, "ACP" for Action Perception and "CT" for Counting.