Learning Compact Video Representations for Efficient Long-form Video Understanding in Large Multimodal Models
Supplementary Material

Yuxiao Chen, Jue Wang, Zhikang Zhang, Jingru Yi, Xu Zhang
Yang Zou, Zhaowei Cai, Jianbo Yuan, Xinyu Li, Hao Yang, Davide Modolo
Amazon AGI

1. Implementation Details

1.1. MLLM Architecture

We use the ViT-G/14 [2] pre-trained in the CLIP style [1, 7] as the visual encoder to extract visual token features from sampled frames. The extracted features are subsequently passed to the pre-trained compressor for feature compression. The compressed features are then mapped into the LLM input space using an 2-layers MLP projector [5]. Following the Llava-OneVision [4], we employ QwenV2-7B [9] as our LLM.

1.2. Adaptive Frame Sampler

In this work, we employ TransNet V2 [8] as the shot boundary detector due to its superior performance and efficiency, enabling real-time processing. Frames are sampled at 10 FPS as input to TransNet V2 [8]. We set the window size for the non-maximum suppression algorithm to 3 seconds.

1.3. Compressor and Decoder Architecture

The design of our compressor primarily follows the architecture of state-of-the-art (Variable-)autoencoders for images [3] and videos [10]. To reduce the parameter size of the compressor, we decompose 3D convolutions into 2D spatial and 1D temporal convolutions, and incorporate bottleneck connections between them.

To be specific, the compressor consists of cascaded *residual blocks*, where each residual block comprises two *convolutional blocks*. Each convolutional block contains a spatial 2D convolutional layer, followed by a 1D temporal convolution and a SiLU activation function. We avoid using 3D convolutions since they introduce a higher number of parameters. The 2D convolutional layer and the 1D temporal convolution are connected in a bottleneck fashion to reduce parameter size: the 2D convolutional layer outputs features with a dimensionality of $d/4$, while the temporal compression maps the features back to dimensionality d, where d represents the dimensionality of the features extracted from the visual encoder. Downsampling is achieved by applying a convolutional stride of 2 in the first convolutional block. Our $4 \times 4 \times 4$ compressor configuration contains three such residual blocks, where the first two blocks perform downsampling, and the overall structure has approximately 65 million parameters. For the decoder, it follows a symmetric architecture to the encoder, with upsampling operations follow those used in VQGAN [3].

Training Statge	Dataset Name	Sample number
Stage 1	ShutterStock	2,827,000
stage 2	ShutterStock	5,113,000
	Ego4d	1,975,861
	BreakFast	20,175
	AVA	183,580
	Vatex	252,200
	SSV2	168,931
	Kinetics	315,709
	total	8,029,456
SFT	ShutterStock	836,750
	NextQA	71,655
	CLEVRER	152,572
	PerceptionTest	7,392
	Ego4d	11,018
	total	1,079,387

Table 1. The composition of training data across different training stages.

1.4. Training Setting

Training Data Mixture: The details of the training data used at different training stages are provided in Table 1. **MLLM training** We train our model on NVIDIA H100 GPUs with a batch size of 512. The training process is divided into three stages. In the stage-1, we train the MLP projector to align the compressed visual tokens with the LLM input space. We keep the LLM, visual encoder, and

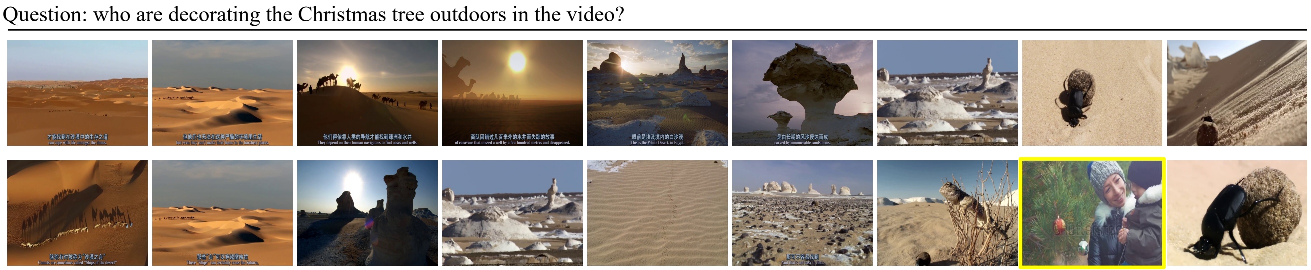
Question: who are decorating the Christmas tree outdoors in the video?

Figure 1. Examples of sampled frames using uniform sampling (top row) compared to our AVS (bottom row). AVS successfully locates the key frame to answer the question.

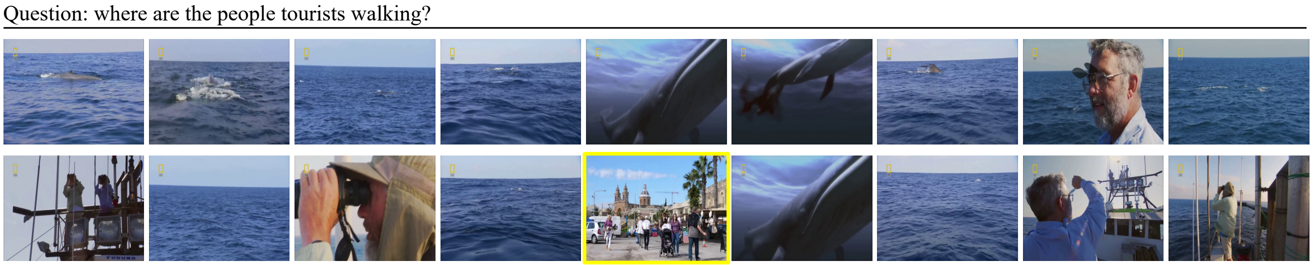

Figure 2. Examples of sampled frames using uniform sampling (top row) compared to our AVS (bottom row). AVS successfully locates the key frame to answer the question.

compressor frozen. The model is optimized by using the AdamW optimizer with a learning rate of 1×10^{-3}. In the stage-2, we train MLLM to learn new visual knowledge. The LLM and the projector are jointly trained with a AdamW optimizer [6] with a learning rate of 1×10^{-5}. In the SFT stage, we fine-tune the MLLM to solve various video understanding tasks and align with human preferences using our SFT data. We train the LLM and the projector with a AdamW optimizer [6] with a learning rate of 1×10^{-5}. *We use 32 frames as input for both training and inference. For the ablation study, the model is trained only during stage 1 and the SFT stage to reduce computational costs. When comparing our method to state-of-the-art approaches (Table 1 in the main paper), the model is trained across stage 1, stage 2, and the SFT stage.*

Compressor pretraining We pre-train our compressor on the Stage 2 training data for one epoch, using a batch size of 512. We use the AdamW optimizer [6] with a learning rate of 2×10^{-5} to optimize the model. Note that the visual encoder is kept frozen.

2. Additional Experiment Results

2.1. Comparison of Video Frame Samplers

In Figure 1 and Figure 2, we additionally present the results of uniform sampling and our Adaptive Video Sampler. Consistent with our observations in the main paper, uniform sampling selects many redundant video frames, inefficiently utilizing the token budget. In contrast, our AVS effectively samples diverse key frames, accurately capturing the frame relevant to the questions "Who is decorating the Christmas tree outdoors in the video?" and "Where are the people tourists walking?" These results further demonstrate the effectiveness of our proposed AVS approach.

References

[1] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 1

[2] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1

[4] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1

[5] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 1

[6] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[8] Tomás Soucek and Jakub Lokoc. Transnet v2: An effective deep network architecture for fast shot transition detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11218–11221, 2024. 1

[9] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 1

[10] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 1