

ORYX MLLM: ON-DEMAND SPATIAL-TEMPORAL UNDERSTANDING AT ARBITRARY RESOLUTION

Anonymous authors

Paper under double-blind review

A GENERATION RESULTS

Video Summarization and Detailed Description. As shown in Fig. 1, the Oryx model effectively generates a comprehensive and detailed caption that accurately summarizes the input video. It captures the main event while preserving essential information.

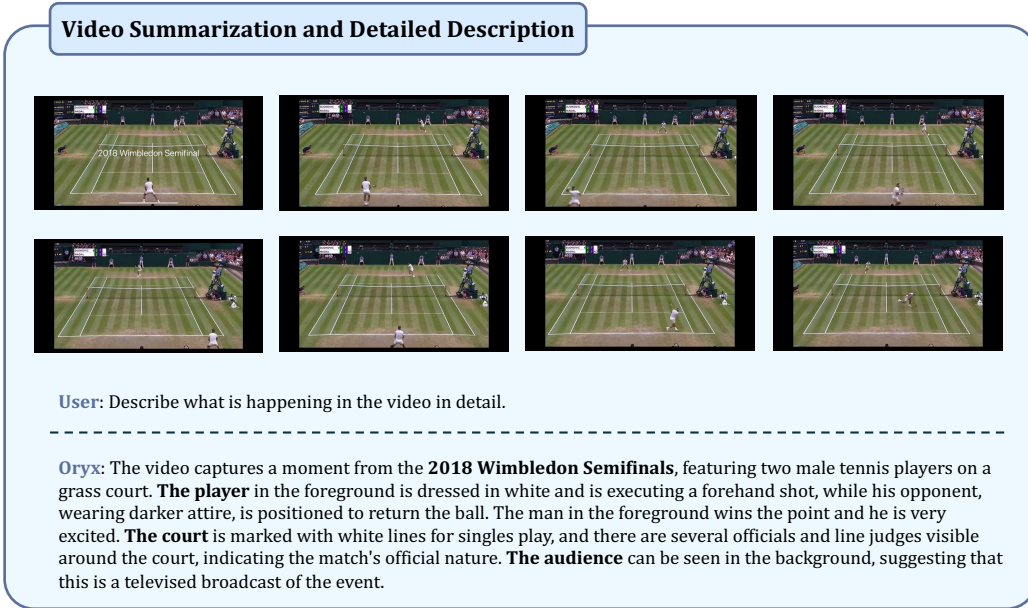


Figure 1: Oryx is able to make a comprehensive video summary and detailed caption.

Video Multiple Choice and Reasoning. Oryx is also capable of reasoning based on the input video. As demonstrated in Fig. 2, Oryx can answer questions through analogy and generate well-reasoned responses.

Skill Learning From Videos. Oryx can acquire useful skills from the input video. As demonstrated in Fig. 3, Oryx learns to use Google Scholar to cite a paper by following the steps shown in the video. It illustrates all the necessary steps to complete the citation, highlighting its strong skill-learning capability and potential for agent-based tasks and task execution.

Understanding 3D with Coarse Correspondences. Oryx enhances its 3D spatial understanding using coarse correspondences. Fig. 4 illustrates Oryx’s reasoning process, demonstrating its ability to improve 3D comprehension through these correspondences and generate accurate reasoning outcomes.

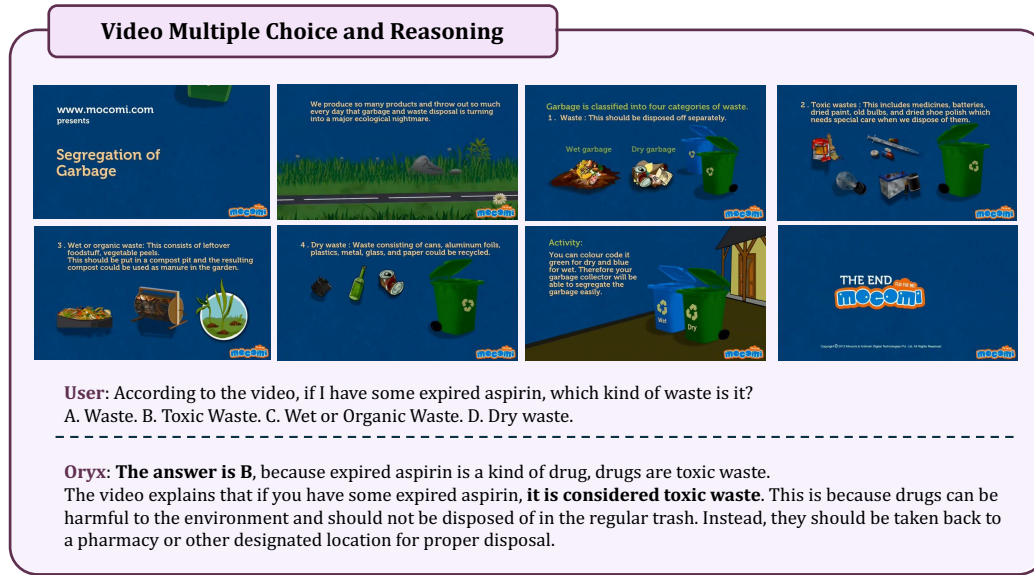


Figure 2: Oryx learns to reason through the input video.

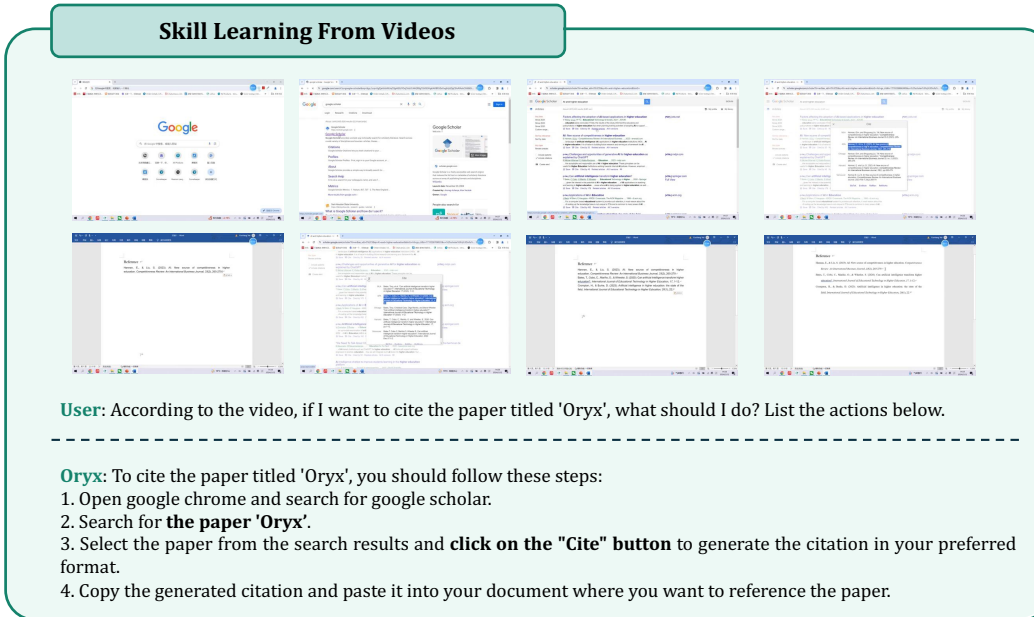
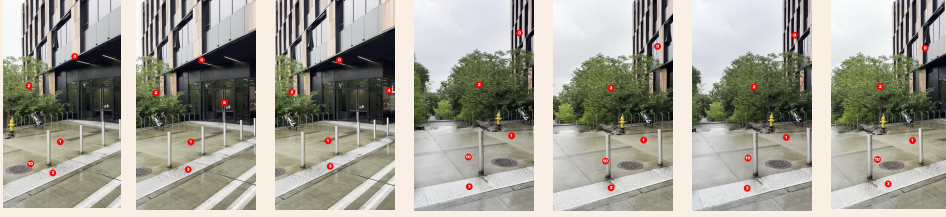


Figure 3: Oryx learns useful skills from the input video.

B VIDEO NEEDLE-IN-A-HAYSTACK

To demonstrate the retrieval ability in long-form visual inputs and test the quality of the dynamic compression module, we design the video needle-in-a-haystack experiment under extreme conditions, following the methodologies established in previous work (Zhang et al., 2024a; Xue et al., 2024). For this experiment, we select an extremely long video and then insert irrelevant image question-answering data as a single frame at arbitrary depths within the video. The model is tasked with answering questions related to these inserted images. We utilize LLaVA-Next-Video (Zhang et al., 2024b) of comparable size as our baseline. As depicted in Figure 5, baseline models trained with 32 frames failed to identify the images, suffering from severe information loss. In contrast, our method successfully retrieves the inserted images and accurately answers the questions, even with

Understanding 3D with Coarse Correspondences



User: If Alice comes out of the building and wants to go down the stairs and cross the street using the pedestrian crossing, which direction should she turn after exiting the door?
A. Left B. Right.

Please answer from Alice's perspective, not the observer's. Explain why using the labels inside the images.

Oryx: If Alice comes out of the building and wants to go down the stairs and cross the street using the pedestrian crossing, she should **turn right after exiting the door**. This is because the pedestrian crossing is located on the right side of the entrance when facing it from inside the building. The labels inside the images indicate that the pedestrian crossing (label 3) is to the right of the door (label 4), and the stairs (label 1) are also situated to the right of the door when viewed from Alice's perspective. Therefore, turning right would lead Alice towards the pedestrian crossing and then across the street.

Figure 4: Oryx understands 3D spatial information through coarse correspondences.

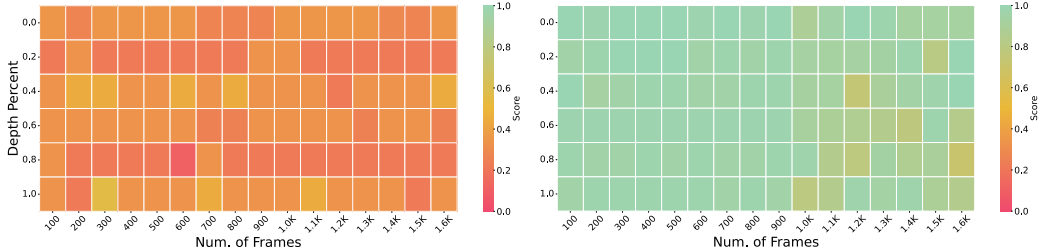


Figure 5: **Visualization Results on Video Needle-In-A-Haystack Experiments.** We compare Oryx-7B (right subfigure) with LLaVA-Next-Video-7B (left subfigure) on the frame retrieval task. The results are shown for inserted depths ranging from 0.0 to 1.0 and the number of frames ranging from 0.1k to 1.6k. The Oryx model demonstrates superior performance in long-form understanding tasks, providing precise results even when a single relevant frame is embedded within over 1k frames of irrelevant information.

frame counts of 1.6k. This outcome strongly demonstrates the model’s ability in long-form temporal understanding, facilitated by the on-demand compression module.

C MORE DETAILS

C.1 IMPLEMENTATION DETAILS

Our implementation integrates the Oryx model with two sets of LLMs, Qwen-2.5-7B (QwenTeam, 2024), and Yi-1.5-34B (Young et al., 2024), to demonstrate generalizability across different model sizes. For the visual encoder, we use our pre-trained OryxViT to support arbitrary-resolution visual inputs. During the pre-training stage, we utilize 558k captioning data from LLaVA-1.5 (Liu et al., 2024), unfreezing the parameters of the dynamic compression module. The image SFT stage involves curating an open-source dataset of around 4M images. In the joint training stage, we incorporate approximately 1.2M data consisting of images sampled from the previous stage and video/3D data. For video data, we restrict the frame number to 64 for standard videos of low compression ratio and 256 for long videos of high compression ratio. We use the 2×2 average downsample for low

compression and 4×4 average downsample for high compression. Image data are maintained at their native resolution, with a maximum size of 1536 pixels, while video data resolutions are confined to a range of 288 to 480 pixels. The rest of the training details are provided in the appendix.

C.2 TRAINING DETAILS

Stage 1. For stage 1, we first pre-train the connector module between the visual encoder and Large Language Model for the initial alignment between image and text modalities. We conduct our experiments on 558k caption data from BLIP (Li et al., 2023) model following LLaVA-1.5 (Liu et al., 2024). We only unfreeze the parameter for the connector while maintaining other parameters fixed. We adopt the total training batch size at 256 and the overall learning rate at $1e-3$. We maintain the aspect ratio for the input image while adjusting the overall pixels to 768^2 to reduce the computational cost. The training cost for the pre-training alignment is lightweight thanks to the small number of parameters for the connector and the relatively lower image-text data pairs. Subsequently, we conduct the supervised fine-tuning stage with 4.1M image data. We freeze the parameter for the visual encoder while unfreezing the connector and the Large Language Model following common practice. In this stage, we use the native resolution of the image while restricting the maximum number of pixels at 1280^2 for efficiency. For the image larger than 1280^2 pixels, we scale down the image to match the overall number of pixels. We set the learning rate at $2e-5$ for Oryx-7B and the learning rate at $1e-5$ for Oryx-34B. We adopt the total batch size at 128 and conduct our experiments on 64 NVIDIA A100-40G GPUs for Oryx-7B and 64 NVIDIA A800-80G GPUs for Oryx-34B, as larger models need more GPU memories. The total model maximum length is set as 8192.

Stage 2. For stage 2, we continuously train the Oryx model from the multi-modal LLMs in stage 1. We randomly sample around 600k image data from the supervised fine-tuning stage in stage 1 and add additional 650k temporal and 3D data from open-source multi-modal datasets, resulting in an overall number of 1.2M further supervised fine-tuning data. In the more general stage, we increase the restriction for image pixels to 1536^2 to meet the longer sequential length in temporal data. We maintain the aspect ratio of video data while normalizing each frame to the minimum size of 288^2 pixels and the maximum size of 480^2 pixels, therefore the token length before compression module ranges from 324 to 900. We adopt 1×1 path for the image data, 2×2 pooling path for the multi-frame data including video and 3D-relevant data, and 4×4 pooling path for the extremely long video needle-in-the-haystack retrieval data. We maintain most of the training hyper-parameters identical to stage 1, with a total batch size of 128, a learning rate of $2e-5$ for Oryx-7B, and a learning rate of $1e-5$ for Oryx-34B. We sample 1 frame per second for video data and set the max frame number at 64 frames. We uniformly sample the frames among all the frames if the number exceeds the upper bound. The maximum sequence length is set to 16384.

D CODE

The code is also provided in the supplementary material (see the `code` folder). Our Oryx is implemented using PyTorch library (Paszke et al., 2019).

REFERENCES

- 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 *ICML*, pp. 19730–19742. PMLR, 2023.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pp. 26296–26306, 2024.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- QwenTeam. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024a.

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, April 2024b. URL <https://llava-vl.github.io/blog/2024-04-30-llava-next-video/>.