Exploring multi-view modality coordination of VLMs' scene understanding

Yunqi (Richard) Gu
Stanford University
yrichard@stanford.edu

Jiacheng (JC) Hu
Stanford University
jchu0822@stanford.edu

1. Introduction

Robotic systems have traditionally been designed to rely heavily on direct sensor inputs and predefined behaviors for tasks like navigation and interaction. These approaches, while effective in controlled settings, often struggle in unstructured environments where adaptability and nuanced contextual understanding are key. The recent advent of wrist-view cameras in robotics signifies a pivotal shift towards enhancing situational awareness through the use of multiple camera angles. This innovation offers a promising avenue for overcoming the limitations of single-camera systems, enabling robots to achieve a more holistic scene comprehension.

Efforts to leverage these multi-view datasets are growing, with the goal of refining a robot's ability to interpret complex environments from multiple video streams. The integration of these visual inputs with linguistic context can potentially transform robotic decision-making processes, making them more aligned with human-like reasoning and interaction. This project aims to investigate the extent to which adding multiple camera perspectives can enhance Vision-Language Models (VLMs) in understanding and interacting with their surroundings. By focusing on the multimodal coordination of visual and verbal cues, we explore new frontiers in robotic cognition.

Given the operational demands of robotic tasks, which often require rapid processing, this study will utilize smaller, more efficient models. These models are chosen for their faster inference times, acknowledging a potential compromise in maximum expressiveness. While this project will not involve real robotic experiments, it will provide valuable insights into the impacts of multi-view camera integration on the performance of VLMs. Through theoretical exploration and computational simulations, we aim to discern whether a more extensive array of camera angles can indeed amplify scene understanding and contribute to smarter robotic behavior. This inquiry not only advances our knowledge in robotic multimodality but also sets the stage for future implementations that could revolu-

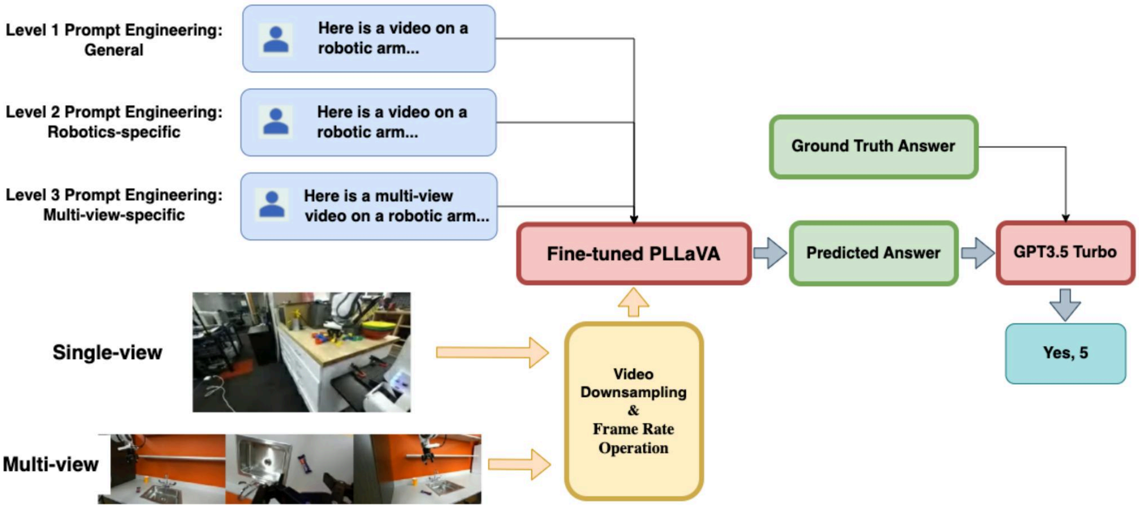

Level 1 Prompt Engineering: General
Here is a video on a robotic arm...
Level 2 Prompt Engineering: Robotics-specific
Here is a video on a robotic arm...
Level 3 Prompt Engineering: Multi-view-specific
Here is a multi-view video on a robotic arm...
Ground Truth Answer
Fine-tuned PLLaVA
Predicted Answer
GPT3.5 Turbo
Yes, 5
Single-view
Multi-view
Video Downsampling & Frame Rate Operation

Figure 1. Illustration of the PLLaVA pipeline

Method	MSRVTT-QA Acc.	ActivityNet-QA Acc.	DROID(single-view) Acc.
PLLaVA 7B	62.0	56.3	35.6

Table 1. Results of video question-answering for PLLaVA 7B

tionize robotic systems in dynamic and unpredictable environments.

2. Related Work

Visually-conditioned language models (VLMs) are a class of models designed to generate or understand language based on visual input. These models combine visual and textual information to perform tasks such as image captioning, visual question answering (VQA), and image-grounded dialogue systems. The goal is to enable the model to comprehend and generate language that accurately describes or interacts with visual content.

A typical VLM architecture [2] usually contains an image processing and visual representations backbone, which could be any visual models by choice. After the image has been processed, it is fed to a multimodality encoder, which is usually either MLPs or some Transformer-based structure, to align with the textual input, and then to any language model by choice along with the prompt. The model architecture is quite modular, and there's a lot of flexibility in choosing which visual and language models works the best for constructing the VLM. Empirical evidence [2] shows that during training, it is advisable to freeze the visual processing layers as well as the LLM while keeping the multimodality encoder open to fine-tuning.

In the context of video understanding, however, traditional VLMs [8] would take exponential time, and the performance might drop due to leveraging on video's noisy representations. The PLLaVA paper by Xu et. al. [9] proposes an approach to scale down this noise when processing videos: embeddings are first learned from videos through CLIP ViT-L [7] and MM projector, yielding visual features with shape (T, w, h, d). These features undergo average pooling, which is applied temporally or spatially. Spatial pooling involves downsampling the spatial dimensions of video frames without significantly degrading performance, with an optimal reduction found at about 50% in PLLaVA paper [9]. This is illustrated by resizing frames to a 12x12 dimension, which was found to balance performance and computational efficiency. Temporal pooling, however, tends to degrade performance as it compresses the temporal dimension, meaning that fewer frames are used to represent time in videos. Overall, the best pooling strategy are found to be $(4, 12, 12, d)$ or $(8, 12, 12, d)$. The pooled features are then flattened and concatenated with question embeddings, serving as input to a language model, similar to VLMs.

3. Dataset

Our experimental investigation will be conducted using the DROID dataset [3], a comprehensive resource tailored originally for the training of robotic manipulation policies. The DROID dataset is characterized by its large scale and in-the-wild deployment, encompassing approximately 76,000 robotic episodes, from which we randomly sampled 20,000 samples of 10-20 seconds long, similar to the training datasets of PLLaVA. Each episode in this dataset is accompanied by one to three language instructions, providing a linguistic context that enriches the visual data captured during the robotic tasks. Moreover, the dataset includes full camera calibrations and video recordings from three distinct camera perspectives—exterior 1, exterior 2, and wrist. These multi-view recordings are pivotal for our analysis, as they offer diverse visual angles on the same robotic actions, thereby facilitating a more robust scene understanding.

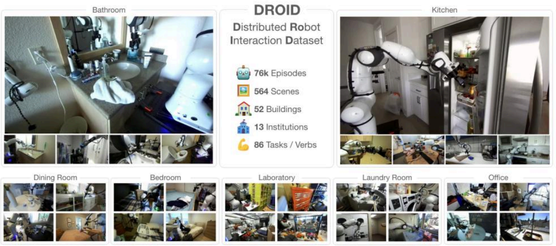


Figure 2. Ilustration of the DROID dataset collected over various camera angles

For the purposes of our study, we will focus primarily on the three different camera perspectives and the accompanying text instructions. These elements will serve as the core inputs for the VLMs employed in our research. Specifically, we will process the videos alongside a predefined prompt (stated in Appendix B) to simulate an interactive scenario where the VLM must interpret and describe the robot's actions. The instructions provided in the dataset will act as ground truth outputs, which will guide the training of our multi-view VLM to align its responses with the intended robotic behavior as described in the instructions.

4. Approach

4.1. Problem Statement

The problem that we try to investigate is quite simple: Given several videos V_1, V_2, ... that describes a robotic scene in multiple different camera angles ($V = \{I_0, I_1, ..., I_T\}$), as well as a constant inquiry promt t_{sample} = 'Describe what this robotic arm is doing', we would like a model M that could output a natural language answer. In addition to the training objective, we design specific questions that further inquire M's capability of demonstrating complex spatial understanding. The experiments will aim to answer the two following questions: would simplifying the robotic video interpretation task to a visually-conditioned language model (VLM) significantly affect model perception? And would incorporating multiple camera views improve the model's spatial reasoning? We will discuss the dataset and our current technical approach in the next following sections.

4.2. Model Structure

To investigate the utility of multi-view video inputs for enhancing robotic video understanding, we propose leveraging three distinct model architectures, reflecting state-of-the-art approaches in video question answering:

First-frame, Last-frame (FFLF) VLM: Due to the specific nature of the DROID dataset (collection of robotic task on video in different camera views), one very simple yet plausible way to make a model understand the task video can be achieved by taking the first frame and the last frame of the videos, "stitch" the two image frames together, pad and resize the stitched images to the required size, then feeding it to a traditional VLM and inquire about the task process. For multiple camera views, we can just all these FFLF images to form a larger image for the VLM to process. The most attractive reason for this approach lies within its simplicity, as we are reducing the video-understanding problem to a structure image understading problem. It will be the baseline method to compare with the other two approaches.

Distinct from the other pipelines that leverages on PLLaVA for video understanding, FFLF is simply a fine-tuned regular VLM. We choose microsoft's **Phi-3 Vision** [1, 5], a comparatively small (4.2B parameter) VLM that yields impressive performance on complex tasks. This model is used to fine-tune the FFLF VLM for scene understanding. We have three cameras recorded in each scene, so our image feeded to the VLM is simply 6 images concatenated together, as illustrated in the attatched image.

Inference with PLLaVA Model: This approach involves downsampling the video to a lower frame rate and resolution, then prompt-engineer the PLLaVA model on the 20,000 random samples of DROID dataset. Given PLLaVA's robust generalizability, we expect to achieve strong zero-shot or few-shot results. As shown in Table 1, without any prompt-engineering or mult-view assist, PLLaVA does not perform too well with the could potentially be improved significantly on DROID dataset with some assists. Due to the concern of computational cost, even if we froze everything but the MM projector, the limited GPU power still didn't allow us to perform fine-tuning on a 7B model

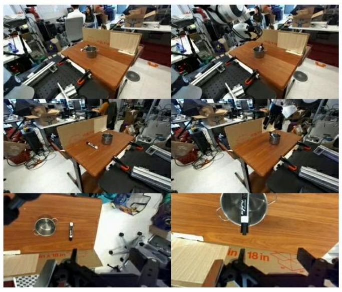


Figure 3. Processed FFLF image data for Phi-3 Vision, with left col. first frames, right col. last frames

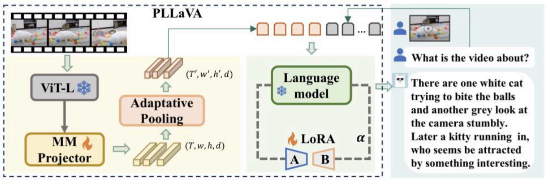

PLLaVA
ViT-L
MM Projector
Adaptative Pooling
(T', w', h', d)
(T, w, h, d)
Language model
LoRA
α
A
B
What is the video about?
There are one white cat trying to bite the balls and another grey look at the camera stumbly. Later a kitty running in, who seems be attracted by something interesting.

Figure 4. PLLaVA: SOTA VLM of Video Question Answering

GPT3.5-Turbo GPT-3.5 Turbo [6] is an advanced AI language model developed by OpenAI, designed for efficient and responsive interactions. We utilize its ability of understanding and comparing two sentences for their semantic information. We ask it to yield a score of matching between 0 to 5, with 5 being the highest, similar to the approach in PLLaVA [9]. We preset scores higher than 3 will be categorized as "Yes." The prompt for evaluation is given in Appendix A.

4.3. Multi-view Input

We will experiment with three different forms of input to understand their impact on the model's performance:

Single-View Video. We will isolate content captured from the *exterior 1* camera angle along with corresponding instructions as shown in Figure 5. This setup serves as a baseline, using a pre-trained VLM fine-tuned on this single-view input.

Sequential Multi-view Inputs will include videos from three camera angles: *exterior 1, exterior 2*, and *wrist*. These are concatenated sequentially in a single row, either in the Wrist-Exterior or Exterior-Wrist order, varying the sequence of these two perspectives. We anticipate that the global pooling layer in PLLaVA, which integrates information across both temporal and spatial dimensions, will

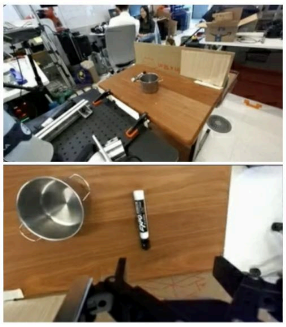


Figure 5. Single view example(top: exterior 1, bottom: wrist)

yield more comprehensive results than analyses based on single-view videos. However, we also expect that the results will be negatively influenced by the length of the concatenated videos, which are 45 seconds long. This duration is significantly longer than the 10-20 second videos used in the training data of PLLaVA, potentially leading to out-of-distribution issues.

Parallel Multi-view The inputs from *exterior 1*, *exterior 2*, and *wrist* are concatenated in parallel and displayed simultaneously, as depicted in Figure 6. This side-by-side presentation of videos is considered the most promising approach: it allows the model to concurrently receive information from different angles, thus avoiding the challenges associated with prolonged video lengths. This method not only enhances the model's ability to integrate diverse visual perspectives but also optimizes the processing efficiency by reducing the temporal burden on video analysis.

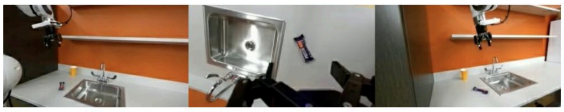


Figure 6. Parallel multi-view example: *Exterior 1*, *Wrist*, *Exterior 2*

Wrist Video + Exterior Image Since data load is an important concern in the field of robotics, we have also tested an approach that requires only one-third the amount of data compared to the two approaches mentioned above. This configuration involves placing video from the wrist camera in the middle, flanked by exterior images on the sides. This method tests the hypothesis that augmenting video input with still images from alternate viewpoints can enrich the model's scene interpretation. It aims to potentially approach the effectiveness of full multi-view video inputs, albeit with some trade-offs in accuracy in exchange for gains in training efficiency.

4.4. Frame Rate

Since PLLaVA operates by pooling the input sequence of encoded images with dimensions (T, H, W, d) into some fixed target dimensions like $4 \times 12 \times 12 \times d$), the number of frames determines how much information is stored in one entry in the pooled result. Therefore, we are interested in how efficient increasing frame rate is for the accuracy.

4.5. Cost Concerns

Simplified Connection Layer. We have opted not to explore the connection layer between video/image encoders and text encoders extensively. Existing research in this area is substantial, and further exploration would likely be cost-prohibitive without yielding significant insights relevant to our specific research query.

Pre-trained Model Weights. We plan to use the smallest available open-weights version of PLLaVA, which has approximately 7 billion parameters from its LLM Vicuna-7b [4], for initial fine-tuning of our video VLM. Unfortunately, even 7B is too much for our computing power, which is 1 H100.

Dataset Optimization. To manage computational resources more effectively, we will truncate the dataset by removing noisy examples, sample $20,000$ from the remaining set, and substantially downsizing each video through less frequent sampling and reduced resolution.

Accumulated Gradient. For training the VLM in the FFLF pipeline, even though Phi-3 Vision is comparatively small, it is still large for a single GPU to handle, so we pass in our data with low batch-size. Instead of updating the model on each batch, we update after enough batches (64 in our implementation) are accumulated to ensure stable training.

These methodological choices are designed to maximize the efficiency of our experiments while exploring the potential benefits of multi-view inputs for enhancing robotic video understanding within the constraints of available technology and resources.

5. Experiments:

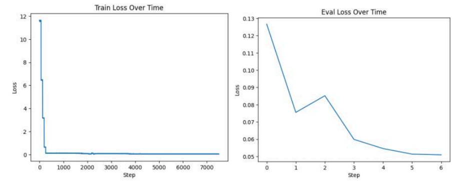
The primary objective of our project is to ascertain whether a multi-view VLM enhances both the accuracy of instruction recovery and the spatial reasoning capabilities in robotic contexts, as compared to a VLM trained with a single camera view. We anticipate that the multi-view VLM will demonstrate superior performance in accurately reconstructing the given instructions due to its exposure to a richer and more varied visual context. This enhancement is expected to stem from the model's ability to integrate diverse visual perspectives, which should theoretically provide a more comprehensive understanding of the scene dynamics and the tasks being performed. Our evaluation strategy consists of both quantitative and qualitative analyses to comprehensively assess the performance enhancements provided by the multi-view VLM.

5.1. FFLF

We include the plot for the training and evaluation loss for the FFLF VLM. Note that the loss for training and evaluation is the automatically-computed cross entropy loss between model's output text distribution versus the ground-truth instruction provided by the dataset. We see that due to accumulated gradients, the train loss shows a step-wise drop after a certain amount of steps and demonstrates sample efficiency during training. Even though the model converges quickly on the training dataset, we see that evaluation loss still decreases over time.

As expected, FFLF doesn't perform very well on robotic video task understanding, at least according to the evaluation pipeline. We have included some samples of our evaluation. In figure 8a, the image-based VLM alone not only understands the setting, but also infers the task performed by the robot given the first and last frames of the video in 3 different angles. It also hints that the fine-tuned vlm has the capability of rationalizing the FFLF structure of the image, as well as incorporating different views to extract information and learn the given scene. In part 8b, the model shows partial understanding of an orange cup and the sink but fails to interpret the action of the robot. The lack of understanding comes from the fact that provided only the first frame and last frame of videos, the model didn't see any shelf in the first place mentioned in the ground truth instruction, and this information loss during problem down-sampling could be interpreted as a form of occlusion. Finally, in figure 8c, our model is unable to interpret what happened in the scene at all. For the actual video, the robotic hand attempted to grab the green block but failed. There's so much information loss during the FFLF process, that even a human being would not interpret the scene correctly given only these images!

Another potential factor that leads to the low performance of the FFLF model may be from our evaluation metric. We discover that GPT-3.5 turbo, the evaluation model we used that outputs a similarity score between two texts, seems to favor longer, more nuanced responses rather than short, concise ones. Since our FFLF model has been fine-tuned on DROID dataset's text instructions, which are usually less than 10 words, it also tends to output short, concise answers, and can potentially lead to low ratings on GPT-3.5 turbo. For instance, for the datapoint illustrated in figure 3, the ground-truth instruction is ”Put the marker in the pot”, while our model outputs ”Put the pen in the cup”. Under the scene's setting, these two instructions are almost identical, but our evaluation marked them different and returns a low similarity score of $2/5$. Upon further investigation into the problem beyond this project, we may select more expressive quantitative evaluators that better represent our task.



(a) Train loss (b) Evaluation loss

Figure 7. Training results on the FFLF VLM pipeline

Some overall great aspects we discovered while testing our fine-tuned FFLF vlm model are: while the model doesn't necessarily understands full robotic action provided just the first and last frames, it demonstrates impressive scene understanding with what it's given, and it is capable of looking for different camera views to find useful information.

5.2. PLLaVA

5.2.1 Multi-view

- The wrist camera offers significantly better results than any exterior cameras. This aligns with the recent findings within the robotics community.

- The improvement that multi-view brings to VLM is best reflected by a parallel presentation of the videos. This is enhanced by a specialized prompting strategy that emphasizes integrating different angles, distinguishing it from single-angle prompts.

- Sequentially concatenating videos is not as effective as the parallel approach. Pllava [9] is trained on videos from 10 seconds to 20 seconds, so 45 seconds of concatenated video might be out of distribution.

- Parallelizing the first frame of the two exterior angles on the sides and the wrist camera video in the middle yields great results. Although slightly less effective than parallelizing three videos directly, it significantly reduces the data load on the robot arm hardware.

5.2.2 Frame-rate

Since PLLaVA operates by pooling the input sequence of encoded images with dimensions (T, H, W, d) into some fixed target dimensions (for example $4 \times 12 \times 12 \times d$), the number of frames determines how much information is stored in one entry in the pooled result. 2x Frame dataset generally performs better than the 1x Frame dataset, as expected.

Input Format	Before Prompt-engineering		After-prompt Engineering	
	Correct	Scores	Correct	Scores
FFLF	9	201	12	229
Exterior-1	21	244	25	284
Exterior 2	28	279	29	283
Wrist	30	288	34	305
Wrist-Exterior Sequential	29	288	49	301
Exterior-Wrist Sequential	26	258	50	335
Parallelly merged	27	275	**55**	**356**
Exterior Image + Wrist Video	26	265	48	342

Table 2. Results of correct/scores for different inputs with/without prompt Engineering with default PLLaVA(100 samples/500 points for score in total)

It's important to note that the experiment on frame rate is crucial, as increasing the frame rate typically enhances outcomes without substantially elevating inference latency. This efficiency is largely due to the role of the pooling layer in PLLaVA, which acts as a bottleneck by aggregating information before it is fed into the Large Language Model (LLM). Since the inference cost associated with the LLM significantly outweighs that of the vision and multimodality encoder, enhancing the frame rate proves to be an effective strategy. This approach allows for more robust data processing without incurring prohibitive computational costs.

5.2.3 Prompting Strategy

Prompting appears to be significantly helping with the prediction. With the same general prompting strategy, which does not specify the key information of each video, PLLaVA performs poorly for all results and does not reflect the improvement of multi-view.

After several rounds of experiments, we discovered several key points for prompting PLLaVA for understanding robotics information:

Video-angle correspondence: Telling PLLava which angle is each video taken from helps with its understanding of the scene.

Key information for each angle: It's also helpful to tell the model which kind of information to look for from each angle. For example, for the two exterior cameras, the environment is very well represented. For the wrist view, the robot's interaction with the object is captured.

Simultaneous start and same scene: We found that it's important to inform the model that the three videos start simultaneously and are about the same view, otherwise it may be confused and consider three videos to be unrelated.

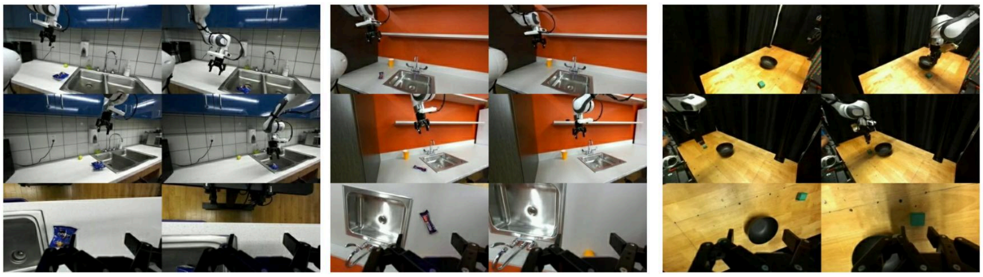
After we prompted it as"For the three images in parallel, the first and third ones are from exterior camera, so you could learn about the environment", the accuracy doubled. Similar phenomenon occurs for sequential concatenation, but it still does not surpass parallel merging. Therefore, prompting is extremely important for multi-view input. For single-view input, the accuracy increases by 20% on average, so despite being less influential than mult-view case, it's still a significant change.

5.3. Inference Time

In the field of robotics, where real-time computing with portable chips is crucial, inference latency is a significant concern. To address this, we explored several time and computation-efficient approaches, such as FFLF and the Exterior Image + Wrist Video approach. We calculated the average inference time for each method and recorded these in Table 4. Notably, the FFLF approach excels due to its image-pipeline nature. In practical scenarios, the significance of scores often surpasses mere accuracy since the signals sent to the robot are continuously refreshed. For instance, a single highly accurate response followed by two markedly incorrect ones is substantially less desirable than three moderately inaccurate responses. This is because the former approach, despite having a 33% accuracy rate, could lead to more significant errors than the latter, which has 0% accuracy. Additionally, we computed a metric known as Score Rate, which represents the model's scoring efficiency per second. Given that the Score Rate for FFLF is considerably higher than that of other approaches, it emerges as a highly promising method for real-world applications, particularly in scenarios that do not demand high precision. Meanwhile, we observed that the Exterior Image + Wrist Video approach does not significantly reduce inference time but does decrease the frame load by two-thirds for other robot hardware components, compared to other multi-view approaches.

6. Conclusion

To assess the VLM proficiency with robotic instructions and its practical applications in robotics, we evaluated two different pipelines, FFLF and PLLaVA, utilizing various viewpoints, input frame rates, prompting strategies, and



(a) VLM fully understands robotic action
Model: Put the blue packet in the sink
Ground Truth: Place the pack of doritos inside the sink

(b) Understands the scene but not the action
Model: Put the orange cup in the sink
Ground Truth: Put the candy bar on the left side of the first shelf

(c) Completely failed understanding
Model: Move the mouse to the left
Ground Truth: Failed to pick up the green block

Figure 8. Visualization of FFLF VLM output vs. Ground Truth

crucially, the influence of different viewpoints. We found that parallel concatenation of videos from different perspectives consistently delivers superior results. Notably, the FFLF pipeline emerged as the most cost-effective method for real-world robotic applications. Additionally, our findings indicate that prompt engineering and higher frame rates significantly enhance accuracy without necessarily increasing computational demands.

However, due to constraints in computing resources and time, we were unable to explore a broader range of topics. Future research could build on our FFLF approach and refine the sampling strategy. A notable limitation in our current implementation is the routine sampling of only the first and last frames, which may miss critical moments if the robot has not yet commenced its operations or has already completed them. Therefore, sampling a greater number of frames, distributed more evenly across the timeline, could achieve a better balance between accuracy and cost-efficiency.

Another area for future exploration is the fine-tuning of the PLLaVA model [9]. Given that the FFLF approach saw substantial improvements through fine-tuning, we anticipate similar enhancements for PLLaVA with adequate fine-tuning. This continued development could further optimize both the performance and the practical utility of VLM in robotic applications.

Input Format	1x Frames		2x Frames	
	Correct	Scores	Correct	Scores
`Exterior-1`	21	244	25	287
`Exterior 2`	28	279	33	293
`Wrist`	30	288	36	304
`Wrist-Exterior Sequential`	29	288	38	310
`Exterior-Wrist Sequential`	26	258	32	290
`Parallelly merged`	27	275	**40**	**335**
`Exterior Image + Wrist Video`	26	265	36	315

Table 3. Results of correct/scores for different inputs with/without speeding up the raw video

Approach	Averaged Inference Time (s)	Score Rate
FFLF	0.0325	7046
Single-view	1.89	161
Multi-view Parallel 2	2.05	173
Multi-view Sequential 4	2.13	157
Exterior Image + Wrist Video	1.95	175

Table 4. Inference time and score rate for each approach

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou.

Phi-3 technical report: A highly capable language model locally on your phone, 2024. 3

[2] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models, 2024. 2

[3] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2024. 2

[4] lmsys. Vicuna7b v1.5. HuggingFace API, 2024. Available from https://huggingface.co/lmsys/vicuna-7b-v1.5. 4

[5] Microsoft. microsoft/phi-3-vision-128k-instruct. HuggingFace API, 2024. Available from https://huggingface.co/microsoft/Phi-3-vision-128k-instruct. 3

[6] OpenAI. Gpt-3.5-turbo. OpenAI's API, 2023. Available from OpenAI: https://platform.openai.com. 3

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[8] Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. Vlm: Task-agnostic video-language model pre-training for video understanding, 2021. 2

[9] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava : Parameter-free llava extension from images to videos for video dense captioning, 2024. 2, 3, 5, 7

A. Evaluation Prompts

We develop our prompt based on the PLLaVA evaluation prompts.

System Prompt: "You are an intelligent chatbot designed for evaluating the correctness of generative outputs for question-answer pairs in the field of robotics. Your task is to compare the predicted answer with the correct answer and determine if they match meaningfully. Here's how you can accomplish the task: — ##INSTRUCTIONS: - Focus on the meaningful match between the predicted answer and the correct answer. - Consider synonyms or paraphrases as valid matches. - Evaluate the correctness of the prediction compared to the answer."

User Prompt: "You will see a video with a robotic arm interacting with its surrounding environment. Please evaluate the following robitic-video-based question-answer pair:
Question: question
Correct Answer: answer
Predicted Answer: pred
Provide your evaluation only as a yes/no and score where the score is an integer value between 0 and 5, with 5 indicating the highest meaningful match. Please generate the response in the form of a Python dictionary string with keys 'pred' and 'score', where value of 'pred' is a string of 'yes' or 'no' and value of 'score' is in INTEGER, not STRING. DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the Python dictionary string. For example, your response should look like this: 'pred': 'yes', 'score': 4.8."

B. Inference Prompts

Original Prompt: The input consists of a sequence of key frames from a video. Answer the question concisely first and followed by significant events, characters, or objects that appear throughout the frames.

Improved Prompt for multi-view parallel parallel: The input consists of three videos from different angles concatenated. The angle in the middle is a wrist camera, which offers view about the object that the robot will operate on. On the sides are exterior cameras, which capture the environment of the robotic arm. They records the same scene and start at the same time.

Improved Prompt for multi-view sequential: The input consists of three videos from different angles concatenated sequentially. The first video occuring is from wrist camera, so you should pay attention to the object the robotic arm is interacting with. The second video and third video are from exterior cameras, so you could learn about the environment.