Vision Language Model For Binary Success Classification of Robot Demonstrations

Yingxin Yao
University of Toronto
maggieyyx.yao@mail.utoronto.ca

Tingji Zhao
University of Toronto
tingji.zhao@mail.utoronto.ca

Ismail Ouazzani Chahdi
University of Toronto
ismail.ouazzani@mail.utoronto.ca

Abstract

Automating the detection of successful robot demonstrations is an opportunity to accelerate the development cycle of machine learning models deployed to physical robots. In this study, we investigate binary success classification by benchmarking and fine-tuning large Vision-Language Models (VLMs) from the InternVL2 family on the Droid dataset, which includes robot demonstrations across diverse tasks and environments. Specifically, we propose a new classification task and processing method to achieve an accuracy of 90% with InternVL2-1B, outperforming standard practices and baseline models such as GPT-4o-mini. These results highlight the effectiveness of incorporating spatial and temporal context for success detection and demonstrate the potential of VLMs as robust tools for robotic evaluation, paving the way for faster advancements in robotics applications.

1 Introduction

Machine learning models are increasingly being deployed in robots operating within the physical world. Unlike controlled simulation environments, the absence of a definitive ground truth signal to indicate the success or failure of robotic actions introduces significant challenges in evaluation and optimization. In such scenarios, automating aspects of data processing and decision-making becomes crucial, as it directly reduces operational costs and enhances scalability. Conducting human annotations after the fact introduces systematic delays in the development cycles of robotic models, as the process requires waiting for annotators to review and label the data.

One approach to addressing this challenge is to develop a task-agnostic system capable of labelling the success or failure of robot demonstrations using affordable, off-the-shelf hardware, such as a single camera. In addition to the ability to rigorously compare how different models perform on the same robot embodiment, this system could allow the automated creation of filtered datasets by rejecting failures for further training of these models.

Our contributions include:

- **Benchmarking Vision-Language Models (VLMs) for Robot Success Classification**: We fine-tune then evaluate the performance of multiple VLM model sizes on success detection of robot demonstrations from a single video.
- **New Classification Task**: We introduce a new approach to process video demonstrations which outperforms the standard method for this application.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Related Works

2.1 InternVL

InternVL, a multimodal large language model (MLLM), is designed to integrate visual and linguistic data for tasks such as visual question answering, grounding, and multimodal reasoning. It combines a Vision Transformer (ViT) with a language model to achieve state-of-the-art performance across various benchmarks, including object detection and multimodal QA tasks [1]. We chose to incorporate InternVL into our system due to its adaptability through fine-tuning, which makes it well-suited for domain-specific applications [1]. Its ability to align visual understanding with contextual linguistic interpretation enables precise and efficient assessment of robotic task completion.

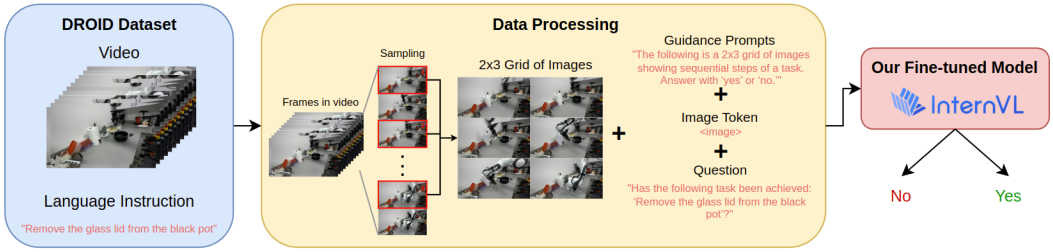
2.2 Success Detection using VLMs

Du et al. [2] introduced the SuccessVQA framework, which formulates success detection as a Visual Question Answering (VQA) task. By fine-tuning three Flamingo 3B vision-language models on three datasets—simulated interactive environments, real-world robotic manipulation, and egocentric human videos—the authors demonstrated the framework's ability to generalize to unseen language and visual conditions. However, the real-world robotic manipulation dataset only contains 2 manipulation tasks in a specific environment, so it is unclear how well it generalizes to different tasks and environments.

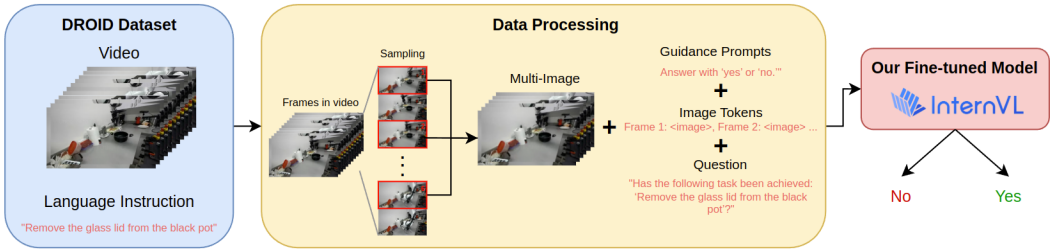
Du et al. [2] proposed the AHA framework, which re-frames failure detection and reasoning in robotic manipulation as a free-form reasoning task rather than a binary classification problem. It not only detects failures but also generates detailed language-based explanations, enhancing its integration into downstream robotic applications like reinforcement learning and task planning.

3 Method

We leverage the Droid dataset to fine tune and evaluate vision-language models for binary success classification of robotic demonstrations. We design and implement two classification tasks—Image Grid and Multi-Image—to explore how video data can be effectively utilized for this purpose. Our code, Docker image and fine-tuning scripts are available on our GitHub.



(a) Image Grid classification task



(b) Multi-Image classification task

Figure 1: System Diagram of the Two Classification Tasks. The system classifies robotic demonstration success or failure using either a grid of video frames or individual frames combined into a single prompt for the vision-language model.

3.1 Droid Dataset

To ensure our model is robust across diverse linguistic and visual contexts, we leveraged the Droid dataset [6]. This dataset encompasses 76,000 robotic manipulation demonstrations spanning 86 distinct tasks and 564 unique environments. Each demonstration includes a language instruction describing the intended manipulation, synchronized video captures from three camera views (one on the robot's wrist and two external), and supplementary robot sensor data.

For our study, we selected the demonstrations with available language instructions at the time of experimentation. From that set, we extracted 13,984 demonstrations that included at least one language instruction. We then divided these into a training set of 11,325 demonstrations (81%) and a validation set of 2,659 demonstrations (19%). Each demonstration was originally labelled as a success, providing a positive example for the training process. To introduce negative examples, we randomly paired each demonstration's video frames with an instruction from a different task. This procedure yielded a balanced set of positive and negative examples, ensuring that the model could learn to distinguish correct task outcomes from incorrect ones.

The Droid dataset provides multiple linguistic variants of each instruction, exposing the model to diverse phrasing and vocabulary. Each demonstration yields multiple data points, with up to three camera views and one to three alternate per demonstration, improving the model's generalization across language styles and perspectives.

3.2 Classification Tasks

We produce datasets for two binary classification tasks using datasets of 170,034 training and 39,762 validation samples from our curated Droid dataset.

Image Grid: Our first approach to input video data into the model is to concatenate sampled frames into an image grid (Figure 1a) . Kim et al. [7] have demonstrated that image grids are a powerful method to encode video information for Visual Question-answering tasks. Following their recommendations, we use a 3x2 grid of uniformly sampled frames from each demonstration, arranged in temporal order from left to right. Both Du et al. [2] and Duan et al. [3] employed image grids for classification tasks. However, Duan et al. [3] used fully horizontal grids, which Kim et al. [7] identified as suboptimal. Meanwhile, Du et al. [2] utilized a large grid combining frames from multiple cameras, reducing resolution and performance—a tradeoff also noted by Kim et al. [7], which may explain their binary classification accuracy of 70% only.

Multi-Image: Our second approach is to process each image individually through the visual encoder and intersperse its tokens with text tokens (Figure 1b) . For example, given two images, one might use the prompt: `"Image 1: <image>, Image 2: <image>. Describe the differences between these two images,"` where the model inserts tokens from the vision encoder into the `"<image>"` placeholders. For consistency, we sample the same six frames as in the Image Grid approach.

Prompt design: We adopted the structured prompt template proposed by Kim et al. [7] as outlined in Figure 1, with three components: (1) a grid reasoning component, specific to the image grid task, (2) task guidance, directing the model to respond in a specific format and (3) the task-related question.

3.3 Fine-tuning with Low-Rank Adaptation

Low-Rank Adaptation (LoRA) [5] fine-tunes pre-trained models by adding low-rank matrices to specific layers while freezing original weights, reducing trainable parameters and therefore the risk overfitting to smaller datasets. Using LoRA, we fine-tuned only 8.8M parameters (1.37%) for InternVL2-1B and 15.7M parameters (0.82%) for InternVL2-2B. We adopt the fine-tuning hyperparameters recommended by InternVL [1], specifics are available on our GitHub repository

4 Experiments

In this section, we evaluate our models quantitatively and qualitatively, comparing pre-trained versions with InternVL2-1B and InternVL2-2B models fine-tuned on Image Grid, Multi-Image or both using LoRA. We also include an external baseline comparison with OpenAI's `gpt-4o-mini`.

Table 1: Validation accuracy comparisons for different models and fine-tuning types.

Model	Fine Tuning	Image Grid Accuracy	Multi-Image Accuracy
`gpt-4o-mini`	None	0.542	0.649
`InternVL2-1B`	Image Grid	**0.8429**	0.859
	Multi-Image	0.5000	**0.899**
	Both	0.499	0.5034
	None	0.612	0.4995
`InternVL2-2B`	Image Grid	0.4963	0.4998
	Multi-Image	0.500	0.4986
	None	0.5003	0.4968

Classification Accuracy: Table 1 summarizes the classification accuracy of each model across various training regimes and datasets. Our best model, InternVL2-1B fine-tuned on Multi-Image, achieved validation accuracy 0.899. In comparison, the baseline InternVL2-1B performed poorly, with 0.612 on Image Grid and 0.4995 on Multi-Image. Interestingly, joint fine-tuning on both Image Grid and Multi-Image tasks reduced accuracy to near random guessing, suggesting interference between datasets. This highlights the need for more advanced models capable of multi-task learning strategies. Moreover, increasing model size did not improve accuracy, implying that the bottleneck lies in fine-tuning or dataset size rather than model capacity, as larger models are more prone to overfitting and poor generalization.

Generalization by Crossing Tasks: The InternVL2-1B model fine-tuned on the Image Grid achieved high accuracy on both the Image Grid and Multi-Image validation sets. Fine-tuning on compact representations enables the model to generalize to less compact ones, but not vice versa: the model fine-tuned on Multi-Image data performed well within its domain but showed baseline-level results on the Image Grid.

Computational Efficiency and Inference Speed: The complexity of the input representation significantly impacts inference speed. The Image Grid approach processes a single fused image with a fixed resolution, enabling relatively faster processing, while the Multi-Image approach processes multiple images independently through the visual encoder, resulting in higher computational overhead. We observed that both training and inference on the Multi-Image dataset was approximately 2–3$\times$ slower than on the Image Grid format on high performance GPUs (A100), reflecting the increased computational cost of handling multiple individual image inputs.

5 Conclusion

In this work, we explored the use of VLMs, specifically InternVL1B and 2B, for binary success classification of robotic demonstrations videos. Through experiments on the Droid dataset, we evaluated the performance of these models across various configurations, demonstrating that multi-image fine-tuning achieved the highest accuracy (90% with InternVL2-1B). These findings underscore the potential of VLMs as robust tools for evaluating robotic systems.

While our current model has demonstrated strong performance in binary success classification for robotic tasks, certain limitations remain unexplored. For example, studying how a single model can generalize across different robotic embodiments such as humanoids or drones remains. Investigating these areas using expanded or augmented datasets could provide further insights into the framework's adaptability to diverse systems.

Several features beyond the scope of this work could significantly expand the model's functionality. For example, the framework currently lacks temporal localization, which involves identifying specific frames where success or failure occurs. This capability would allow for a more detailed analysis of robotic performance and could serve as a valuable diagnostic tool. Future research could draw on benchmarks like the Ego4D dataset [4], which provides tools and evaluations for temporally grounding actions within video sequences, to address this challenge. These extensions represent promising avenues for enhancing the robustness and utility of the model in the robotic development cycle.

References

[1] Zhe Chen et al. *How Far Are We to GPT-4V? Closing the Gap to Commercial Multimodal Models with Open-Source Suites*. 2024. arXiv: 2404.16821 [cs.CV]. URL: https://arxiv.org/abs/2404.16821.

[2] Yuqing Du et al. "Vision-Language Models as Success Detectors". In: arXiv:2303.07280 (Mar. 2023). arXiv:2303.07280. URL: http://arxiv.org/abs/2303.07280.

[3] Jiafei Duan et al. *AHA: A Vision-Language-Model for Detecting and Reasoning Over Failures in Robotic Manipulation*. en. Oct. 2024. URL: https://arxiv.org/abs/2410.00371v1.

[4] Kristen Grauman et al. *Ego4D: Around the World in 3,000 Hours of Egocentric Video*. 2022. arXiv: 2110.07058 [cs.CV]. URL: https://arxiv.org/abs/2110.07058.

[5] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. en. June 2021. URL: https://arxiv.org/abs/2106.09685v2.

[6] Alexander Khazatsky et al. "Droid: A large-scale in-the-wild robot manipulation dataset". In: *arXiv preprint arXiv:2403.12945* (2024).

[7] Wonkyun Kim et al. *An Image Grid Can Be Worth a Video: Zero-shot Video Question Answering Using a VLM*. 2024. arXiv: 2403.18406 [cs.CV]. URL: https://arxiv.org/abs/2403.18406.

Individual Contributions

Yingxin Yao

- Model selection.
- Data processing code.
- System Flow Diagram.
- Tables and Figures.
- Report writing.

David Zhao

- Model selection.
- Interfacing with openai model.
- Benchmarking code.
- Benchmarking baseline and fine-tuned models.
- Discussing Experimental result
- Report writing.

Ismail Ouazzani Chahdi

- Dataset selection.
- Data processing code.
- Generation of datasets.
- Dockerization of InternVL dependencies for finne-tuning.
- Fine-tuning scripts.
- Fine-tuning the models.
- Report writing.