Vision-Language Models for Robot Success Detection

Fiona Luo

General Robotics, Automation, Sensing and Perception (GRASP) Laboratory,
University of Pennsylvania, 3330 Walnut St, Philadelphia, PA 19104
fionaluo@seas.upenn.edu

Abstract

In this work, we use Vision-Language Models (VLMs) as a binary success detector given a robot observation and task description, formulated as a Visual Question Answering (VQA) problem. We fine-tune the open-source MiniGPT-4 VLM to detect success on robot trajectories from the Berkeley Bridge and Berkeley AUTOLab UR5 datasets. We find that while a handful of test distribution trajectories can train an accurate detector, transferring learning between different environments is challenging due to distribution shift. In addition, while our VLM is robust to language variations, it is less robust to visual variations. In the future, more powerful VLMs such as Gemini and GPT-4 have the potential to be more accurate and robust success detectors, and success detectors can provide a sparse binary reward to improve existing policies.

Introduction

Detecting task success is an important problem for several reasons. Success detection ensures task completion and allows agents to recognize and recover from errors. It aids robots in fine-tuning by generating a surrogate reward, allowing policy improvement with minimal human intervention (Yang et al. 2023). Further, automated success detection reduces efforts spent on manually annotating robot data.

Generalizability is an important consideration since a success detector should be able to annotate diverse robot observations. A success detector should generalize given the following alterations.

1. *Visual/environment variation.* Images indicating success on the same task can have a different viewpoint, lighting, background, environment setup, and distractor objects.
2. *Language variation.* Task descriptions which represent the same goal can differ semantically, use synonyms, or have varying levels of specificity.
3. *Task Generalization*. A success detector should generalize to unseen tasks, given that the model has already seen a large number of tasks in similar environments.

VLMs are then a natural choice for a robust success detector because they can be pre-trained on internet scale image-text pairs. We use MiniGPT-4 because it is open-source and can be fine-tuned with in-distribution robot data.

Contributions

This project makes the following contributions.

- *Accurate Few-Shot Success Detector*. MiniGPT-4 fine-tuned with test-distribution trajectories achieves $\geq 95\%$ balanced accuracy on two Berkeley UR5 tasks.
- *Analysis of Cross-Dataset Influence*. We investigate how fine-tuning on one robot dataset affects performance on a different dataset.
- *Visual and Language Variation Study*. We find that while accuracy decreases for visual variations, our success detector is highly robust to language variations.

Related Work

Vision-Language Models VLMs aim to understand and generate textual information by jointly processing visual and textual data. Examples include CLIP and Flamingo, which are trained with contrastive objectives (Radford et al. 2021) (Alayrac et al. 2022). Current state-of-the-art VLMs include GPT-4 and Gemini (OpenAI 2023) (Akter et al. 2023). In this work, we use MiniGPT-4 which aligns a frozen visual encoder with a frozen LLM (Zhu et al. 2023).

VLMs for Success Detection Most similar to our work is RoboFUME, which also uses MiniGPT-4 as a success detector fine-tuned with the Berkeley Bridge dataset (Yang et al. 2023). Our research builds upon RoboFUME by examining the low data regime, effects of co-finetuning, and the visual and language robustness of the model. Also closely related is DeepMind's use of Flamingo as a binary success detector on robot observations given VQA prompts (Du et al. 2023). In similar spirit, works such as VIP and LIV seek to encode a universal reward function from vision-language representations (Ma et al. 2023b) (Ma et al. 2023a).

Approach

Datasets We use the Berkeley Bridge dataset and Berkeley UR5 Demonstration Dataset (Walke et al. 2023) (Chen, Adebola, and Goldberg 2023). Example demonstrations are shown in Figure 1. The Bridge dataset includes 60,096 trajectories with 13 skills and 24 environments, including kitchens, tabletops, sinks, and laundry machines. The Berkeley UR5 Demonstration Dataset has 896 trajectories and 4 tasks on the same tabletop environment.

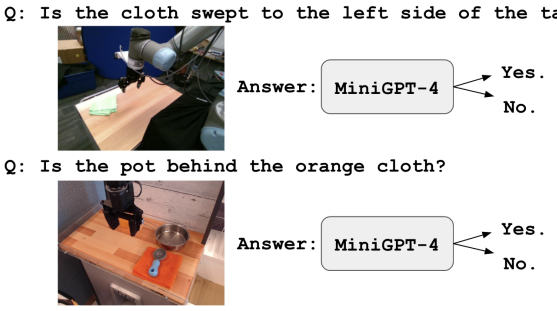

Figure 1. Success detection as a VQA task using demonstrations from the UR5 (top) and Bridge (bottom) datasets.

To classify images as success or failure, we assign the last 3 images of a trajectory to be successful and randomly choose 5 images from all but the last 10 images of the trajectory to be failures, similar to RoboFUME (Yang et al. 2023). We rephrase each task instruction into a question and the answer to each prompt is "Yes." or "No.", a binary response.

MiniGPT-4 Fine-tuning We use the language model Vicuna-13B with MiniGPT-4 (Chiang et al. 2023). MiniGPT-4 is pre-trained on over 5 million image-text pairs. We fine-tune the pre-trained checkpoint using observations from 5 to 400 trajectories per dataset.

Experimental Results

Few-shot Success Detection on UR5 Dataset We build a few-shot success detector on two UR5 tasks: "Sweep the green cloth to the left side of the table" and "Pick up the blue cup and put it into the brown cup." We evaluate on 100 unseen observations. As shown in Tables 1 and 2, as little as 5 trajectories can achieve a balanced accuracy of 92% and 94% respectively. Without fine-tuning, zero-shot accuracy is poor around 50% in part because the model has trouble outputting a binary response. False positive rates are low after fine-tuning which is ideal to avoid reward exploitation.

# trajectories	Balanced Accuracy	FPR	FNR	Precision
0	49%	30%	73%	22%
5	92%	2%	14%	97%
10	86%	**0%**	28%	**100%**
25	**95%**	2%	**8%**	97%

Table 1. Accuracy on UR5 cloth sweeping versus amount of fine-tuning trajectories

# trajectories	Balanced Accuracy	FPR	FNR	Precision
0	56%	71%	16%	51%
5	94%	**2%**	9%	**96%**
10	91%	5%	13%	91%
25	**97%**	7%	**0%**	90%

Table 2. Accuracy on UR5 cup stacking versus amount of fine-tuning trajectories

Fine-tuning on Berkeley Bridge Dataset We fine-tune with 400 trajectories from the Bridge Dataset. The majority of tasks are unique although objects and skills are recycled. We evaluate using 25 Bridge trajectories of unseen tasks.

Results are shown in Table 3. We achieve a balanced train accuracy of 87.6% and a test accuracy of 70.3%. Though lower, the test accuracy reveals that MiniGPT-4 is able to generalize to unseen tasks if it has familiarity with the environment and objects. In contrast, when we fine-tuned MiniGPT-4 on two UR5 tasks and tested on two unseen UR5 tasks, the VLM was not able to generalize, indicating that just one or two skills is not sufficient for generalization.

	Balanced Accuracy	FPR	FNR	Precision
Train	87.6%	15.6%	9.2%	85.3%
Test	70.3%	33.3%	26.0%	69.2%

Table 3. Train and Test Accuracy on Bridge Dataset

Co-finetuning Study We study whether co-finetuning with the Bridge and UR5 datasets can improve accuracy on the UR5 dataset. Ultimately as shown in table 4, co-finetuning has lower accuracy as the two dataset distributions are too different. Fine-tuning on solely Bridge causes the VLM to predict majority failures on the UR5 dataset. Co-finetuning performs worse than fine-tuning with only UR5, as the model optimizes to the less relevant Bridge dataset.

# trajectories	BA	FPR	FNR	Precision
25 UR5, 0 Bridge	95%	2%	8%	97%
0 UR5 400 Bridge	52%	2%	94%	63%
25 UR5, 400 Bridge	90%	3%	17%	94%

Table 4. Accuracy on UR5 cloth sweep with co-finetuning

Robustness to Visual and Language Variations To test visual robustness, we apply 10 augmentations to a test dataset of 25 UR5 cloth sweeping trajectories including rotation, brightness, contrast, crop, sharpen, blur, noise, and adding shapes. To study language robustness, we paraphrase the cloth sweeping prompt in 8 different ways that change semantic structure and use synonyms. Results in Table 5 show that balanced accuracy decreases by 20% with visual augmentations but only 1% with paraphrasing. Thus, we are likely bottlenecked by the vision and not language model.

	Balanced Accuracy	FPR	FNR	Precision
Original	95%	2%	8%	97%
Augmented	75%	0%	51%	100%
Paraphrased	94%	0%	12%	100%

Table 5. Accuracy on UR5 cloth with augmentations and paraphrasing

Conclusion

We develop an accurate few-shot success detector using the VLM MiniGPT-4. When fine-tuned on Berkeley Bridge, the detector can also generalize to unseen Bridge tasks. In the future, extending this work to more powerful VLMs such as GPT-4 or Gemini zero-shot is promising direction, as well as exploring how success can be used as a sparse reward.

Acknowledgments

This research was conducted as part of Professor Dinesh Jayaraman's Vision for Robot Learning course at the University of Pennsylvania. We are grateful towards Jason Ma and Edward Hu for their valuable advice and insights throughout the project.

References

Akter, S. N.; Yu, Z.; Muhamed, A.; Ou, T.; Bäuerle, A.; Ángel Alexander Cabrera; Dholakia, K.; Xiong, C.; and Neubig, G. 2023. An In-depth Look at Gemini's Language Abilities. arXiv:2312.11444.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; Ring, R.; Rutherford, E.; Cabi, S.; Han, T.; Gong, Z.; Samangooei, S.; Monteiro, M.; Menick, J.; Borgeaud, S.; Brock, A.; Nematzadeh, A.; Sharifzadeh, S.; Binkowski, M.; Barreira, R.; Vinyals, O.; Zisserman, A.; and Simonyan, K. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. arXiv:2204.14198.

Chen, L. Y.; Adebola, S.; and Goldberg, K. 2023. Berkeley UR5 Demonstration Dataset. https://sites.google.com/view/berkeley-ur5/home.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Du, Y.; Konyushkova, K.; Denil, M.; Raju, A.; Landon, J.; Hill, F.; de Freitas, N.; and Cabi, S. 2023. Vision-Language Models as Success Detectors. arXiv:2303.07280.

Ma, Y. J.; Liang, W.; Som, V.; Kumar, V.; Zhang, A.; Bastani, O.; and Jayaraman, D. 2023a. LIV: Language-Image Representations and Rewards for Robotic Control. arXiv:2306.00958.

Ma, Y. J.; Sodhani, S.; Jayaraman, D.; Bastani, O.; Kumar, V.; and Zhang, A. 2023b. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. arXiv:2210.00030.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Walke, H.; Black, K.; Lee, A.; Kim, M. J.; Du, M.; Zheng, C.; Zhao, T.; Hansen-Estruch, P.; Vuong, Q.; He, A.; Myers, V.; Fang, K.; Finn, C.; and Levine, S. 2023. BridgeData V2: A Dataset for Robot Learning at Scale. arXiv:2308.12952.

Yang, J.; Mark, M. S.; Vu, B.; Sharma, A.; Bohg, J.; and Finn, C. 2023. Robot Fine-Tuning Made Easy: Pre-Training Rewards and Policies for Autonomous Real-World Reinforcement Learning. arXiv:2310.15145.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. arXiv:2304.10592.