Bottlenecks of Vision Language Models in Image to PDDL State Extraction

Gaëlic Bechu, Ehsan Abbasnejad, Mihai Andries, Panagiotis Papadakis

Bottlenecks of Vision Language Models in Image to PDDL State Extraction

Gaëlic BECHU*‡§, Ehsan ABBASNEJAD†§, Mihai ANDRIES* and Panagiotis PAPADAKIS*§
*IMT Atlantique, Lab-STICC, UMR CNRS 6285, team RAMBO, F-29238 Brest, France
†FIT, MONASH University, Melbourne, Australia
‡AIML, University of Adelaide, Adelaide, Australia
§CNRS, IRL 2010 CROSSING Adelaide, Australia

***Abstract*—The advent of Large, Pre-trained, Vision or Language Models (LLM, VLM, etc.) has lead to their wide use in multiple robotic applications. To better understand how they can be leveraged in robotics applications in generating a precise description of an environment from an image, we evaluate the generation of Natural Language (NL) descriptions from images by comparing different prompts, image-to-PDDL data and state-of-the-art Vision-Language Models (VLMs). Results reveal specific strengths and weaknesses of individual VLMs and a strong reliance on image complexity in terms of object shape, cardinality and position, when generating NL descriptions.**

***Index Terms*—VLM, LLM, PDDL**

I. Introduction

The advent of foundation models (LLM, VLM, etc.) led to the development of Vision-Language-Actions Models (VLA) [1]. When given an image of a scene and a textual instruction in natural language, VLAs can compute an appropriate action as output. Unsurprisingly, the quality of the output action plan is highly dependent on the quality of the input scene description, or otherwise put, on the description of the initial state. In this study, we evaluate the ability of pre-trained VLMs to generate accurate state descriptions from images.

To evaluate state of the art models in describing images, we tested open source VLMs (Llava-One [3], CogVLM2 [4], and Eagle2 [5]) related to highly ranked models in the following study [2].The study evaluates VLMs's ability to extract visual features through pairing images and prompts with no correlation to prevent giving information through the prompt and evaluating the description of the image. Our study focuses on state descriptions usable for robotic action planning, through Planning Domain Definition Language (PDDL) [7], standard planning environments (BlocksWorld) and compatible simulators [6].

For benchmarking purposes, we created three datasets of image-label pairs of incremental realism (from simulated to real-world scenes). These datasets contain more than 100 images ranging from simple problems to more complex ones (1-12 objects). We evaluate VLMs' ability to describe scenes by comparing the generated and ground-truth PDDL. Each sample image in our datasets is accompanied by a corresponding NL and symbolic (PDDL) description.

While there have been studies on the extraction of symbolic descriptions from images (using one-shot or few-shot approaches), those typically provide context information in prompts [8], [9]. In contrast, our study evaluates the accuracy of NL descriptions generated without providing context information. On this basis, we experimentally assess the ability of VLMs to generate a NL description containing all necessary information for a successful transformation to a symbolic representation.

II. Performance evaluation framework

A. Problem statement

We focus on the translation of images into the required NL information for the generation of an initial state description by using a VLM. Through a prompt p_r provided to the VLM we seek to an answer s_0 that will contain all the required information for the generation of a PDDL problem file, without inducing hallucination, and in the absence of context (to avoid information leakages). This can be expressed:

$$VLM(p_r, i(\epsilon)) = s_0 \tag{1}$$

where i is an image taken from an environment ϵ.

Following [10], a finite and discrete state space is defined as S = $<S, s_0, SG, A, f, c>$, to describe a problem in PDDL. The problem file is split into three parts: the object list, the (known) initial state $s_0 \in S$, and the set of goal states $SG \subseteq S$. We focus on generating the object list and the initial state in NL format as they contain the information required for generating the initial state in PDDL format.

B. Evaluation protocol

We designed a set of prompts that were tested across our 3 datasets and different VLMs, following the same prompt design as [8]. These prompts were designed to generate a broad description of the environment while giving no information in the context of the prompt. The purpose of the design was to extract information purely from a vision standpoint.

We categorized the errors in the generated NL across multiple VLMs and prompts, reflecting commonly encountered flaws of VLMs and their impact on the generated problems by increasing order of importance (from 1 to 4). Each NL description may only generate a single type of error. Each error is sorted according to a component of the PDDL problem file (object list and initial state). The object list of the PDDL problem file contains the object types, their number and the initial state contains their states. We also concern ourselves

with the ability of VLM to extract visual features of the objects from the image. We do not evaluate the generation of the goal state SG, because it is usually given in NL.

We arranged the generated descriptions the errors into four priorities in order of importance. Priorities 1 and 2 represent the inability of a VLM to generate a description matching the object list from the image. We split the erroneous NL description into two sub-categories, hallucinations and miscount. The presence of an object in the description that is not present in the image, is considered an error by hallucination (Priority 1), while a description containing the wrong number of objects is classified as a miscount (Priority 2). These errors, which stem from limits of foundation models (hallucinations, counting), are prioritized by their impact.

Priority 3 represents the inability of the VLMs to describe the state of objects matching the initial state s_0 (II-A) in the image. Our datasets consist of images of the blocks-world domain. The actions available in the domain file of the blocks-world domain, where the agent is a single robotic arm, are actions for manipulating the blocks on the table (unstack, stack, etc.). The necessary information for the manipulation of these blocks is their state (position). Priority 4 addresses the inability of VLMs to differentiate between different objects, here represented by different colors.

We created these evaluation criteria to sort through the generated initial states and analyzed the conditions where VLMs exhibit inferior performance. To sort these initial states, we consider this NL description as ground truth (provided with the datasets). We automatically and manually compare the generated NL descriptions to the ground truth.

III. Experiments

The benchmarking datasets that we developed and employed in this section III are available at https://gitlab.imt-atlantique.fr/g21bechu/vlm4robot.git. We first used a single VLM (`Llava-One 7B ov` [3]) to evaluate different prompts over the three datasets, allowing to identify the best performing ones. In particular, we selected five prompts giving the most accurate initial state s_0 across the 3 datasets.

The results suggest that for the blocks-world domain, VLMs are not impacted by hallucinations, and are able to accurately perceive the initial state of the scene (SD:7%, SRWD:56%, RWD:21%). However, their accuracy decreases for a number of objects greater or equal to 6, due to an increased probability of miscounting (Priority 2). The SD dataset contains 50 $\%$ of its images with more than 6 blocks, which lead to 40 $\%$ more miscount errors than the other datasets . We chose the VLM (Llava-One) and dataset (SRWD) pair generating the most accurate descriptions to evaluate the decrease of accuracy related to an increase in complexity.

The color evaluation further reveals performance differences across the datasets, namely, the RWD dataset is the one mostly impacted by errors, followed by SRWD while there are no such errors in SD. The SD dataset has randomly assigned colors to the cubes, differing from the other two datasets with similar colors to the cubes. RWD and SRWD possess the same environment (background object, colors), the main difference is the fact that one dataset consists of simulated images and the other one from real world images. Through analysis of the color evaluation, we concluded that the errors common to both RWD and SRWD occur from mislabeling the colors present in the image as another color (pink as white, purple as pink, etc.).

Based on the performances of the different VLMs, Llava-One is the best one in 2 out of 3 datasets, EAGLE2 is the last one in all datasets, and CogVLM2 takes second place. Llava-One has much better results (20%) on SRWD and CogVLM2 (10%) on RWD. These results suggest that some VLMs may have been trained to different extents on real data and on synthetic data. Through the disparity across the different VLMs, we conclude that it is required to experiment with VLMs in order to use the one most fitting for each objective. The experimentation requires both resources and time. It will benefit from the generation of a benchmark evaluating VLM geared toward robotic manipulation.

IV. Conclusion

Although pre-trained VLMs can describe the necessary information (number, position and color), the accuracy of these generated NL descriptions decreases as the number of objects increases. It also depends on the VLMs used for the generation as well as the image of the environment (simulation or real). The models used in these experiments are all freely accessible on the Web. The training of the models determines on which type of dataset they will perform better. Furthermore, pre-trained VLMs are only accurate till the complexity of the image becomes too high. The next steps will entail replicating the experiments on different planning domains and models.

References

[1] Wang, Zhijie et al. "VLATest: Testing and Evaluating Vision-Language-Action Models for Robotic Manipulation. " FSE 2025

[2] Zhecan Wang et al. " JourneyBench: A Challenging One-Stop Vision-Language Understanding Benchmark of Generated Images " NeurIPS 2024

[3] Li et al. "LLaVA-OneVision: Easy Visual Task Transfer" CoRR 2024

[4] Wenyi Hong et al. "CogVLM2: Visual Language Models for Image and Video Understanding" Arvix abs/2408.16500 (2024)

[5] Yuhui Li et al., "EAGLE-2: Faster Inference of Language Models with Dynamic Draft Trees" EMNLP, 2024

[6] G. Béchu et al. "A software engineering point of view on digital twin architecture," 2022 ETFA, pp. 1-4

[7] McDermott, Drew et al. "PDDL-the planning domain definition language." (1998). AIPS-98 International Planning Competition

[8] Keisuke Shirai et al. "Vision-Language Interpreter for Robot Task Planning" 2024, (ICRA)

[9] Naoaki Kanazawa et al. "Real-world cooking robot system from recipes based on food state recognition using foundation models and PDDL" 2024, Advanced Robotics, vol 38, p. 1318 - 1334

[10] Michael Katz et al. "Make Planning Research Rigorous Again!" 2025

[11] Emanuele De Pellegrin et al. " Planning Domain Simulation: An Interactive System for Plan Visualisation " 2024, ICAPS