LADEV: A Language-Driven Testing and Evaluation Platform for Vision-Language-Action Models in Robotic Manipulation

Zhijie Wang[1], Zhehua Zhou[1], Jiayang Song[1], Yuheng Huang[2], Zhan Shu[1], and Lei Ma[2,1]

***Abstract*— Building on Large Language Models (LLMs) and Vision Language Models (VLMs), recent research has introduced Vision-Language-Action (VLA) models as an end-to-end solution for robotic manipulation tasks. By taking camera images and natural language task instructions as inputs, VLA models directly determine the control actions required for the robots to complete specified tasks, enhancing both their decision-making abilities and interactions with human users. However, due to the data-driven nature and lack of interpretability, ensuring the effectiveness and robustness of VLA models is challenging. This highlights the need for a reliable testing and evaluation platform. For this purpose, in this work, we propose LADEV, a comprehensive and efficient platform specifically designed for evaluating VLA models. We first present a language-driven approach that automatically generates simulation environments from natural language inputs, mitigating the need for manual adjustments and greatly improving testing efficiency. Then, to further assess the influence of language input, we implement a paraphrase mechanism that produces diverse natural language task instructions for testing. Finally, to expedite the evaluation process, we introduce a batch-style method for conducting large-scale testing of VLA models. Using the proposed LADEV, we examine the performance of multiple state-of-the-art VLA models. We show that our platform not only serves as an efficient tool for evaluating VLA models but also establishes a solid baseline for advancing these models, paving the way for realizing more intelligent and autonomous robotic systems.**

I. Introduction

Recent research has demonstrated the application of Large Language Models (LLMs) in various robotic domains [1], [2], where they are employed to tackle complex tasks that usually require human-like cognitive abilities, such as planning [3]–[5], task comprehension [6]–[8], and intention understanding [9]–[11]. Building on these advancements, a growing number of works now also employs Vision Language Models (VLMs) [12] to enhance robots with the ability to process visual inputs [13]–[15]. It enables robots to interpret their surrounding environments and identify interactable objects, facilitating autonomous decision-making processes necessary for task completion.

This development has led to the emergence of a new class of end-to-end models known as Vision-Language-Action (VLA) models [16], [17], which are predominantly designed for robotic manipulation tasks [18]–[20]. The inputs to the VLA models consist of images captured by cameras and user-provided natural language instructions that describe the desired task. Utilizing these inputs, the VLA models then directly generate control commands, e.g., the pose of the end-effector, to guide the robotic manipulator in completing the assigned tasks [18] (see Fig. 1). Trained on extensive and diverse datasets, VLA models introduce a novel approach to robotic control that not only mitigates the need for programming low-level tasks and motion controllers but also fosters direct interaction between robots and users through natural language instructions. This innovation represents a potentially significant step toward achieving advanced robot intelligence and the eventual realization of fully autonomous and intelligent robotic systems [21].

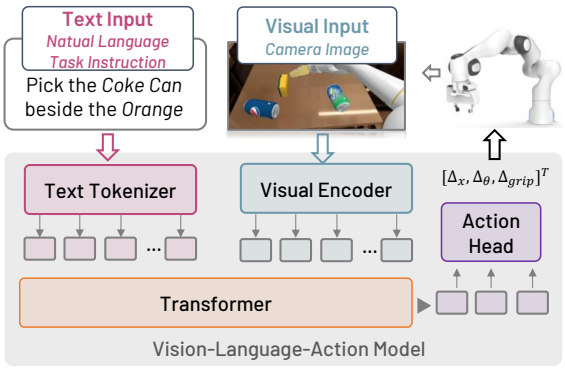

Fig. 1: The VLA model takes camera images and natural language task instructions as inputs. Using a transformer-based encoding and decoding process, the VLA model directly generates control commands for the robots.

However, the data-driven nature of VLA models introduces several challenges. For example, the effectiveness of task execution is highly reliant on the quality of the training data used to develop these models [16]. Moreover, the lack of interpretability in VLA models makes it difficult to guarantee their reliability and robustness [22]. These challenges underscore the need for a comprehensive platform to rigorously test and evaluate the performance of VLA models across a variety of manipulation tasks and scenarios.

Unfortunately, as an early exploratory approach, there is still no platform specifically designed for evaluating VLA models. Recently, a simulation platform called SimplerEnv was introduced in [23]. Built on the SAPIEN simulator [24] and the ManiSkill2 benchmark [25], it includes multiple typical pick-and-place scenarios and various VLA models. By generating simulation environments that replicate real-world training conditions, SimplerEnv is able to assess the performance of VLA models in a simulated setting and provide a summary of the results. However, modifying the

[1]Zhijie Wang, Zhehua Zhou, Jiayang Song, and Zhan Shu are with the University of Alberta, Edmonton, AB, Canada {zhijie.wang, zhehua1, jiayan13, zshu1}@ualberta.ca

[2]Yuheng Huang and Lei Ma are with The University of Tokyo, Tokyo, Japan. Lei Ma is also with the University of Alberta yuhenghuang42@g.ecc.u-tokyo.ac.jp, ma.lei@acm.org. The code of this paper will be made available online soon. Additional information and materials are provided in https://sites.google.com/view/ladev.

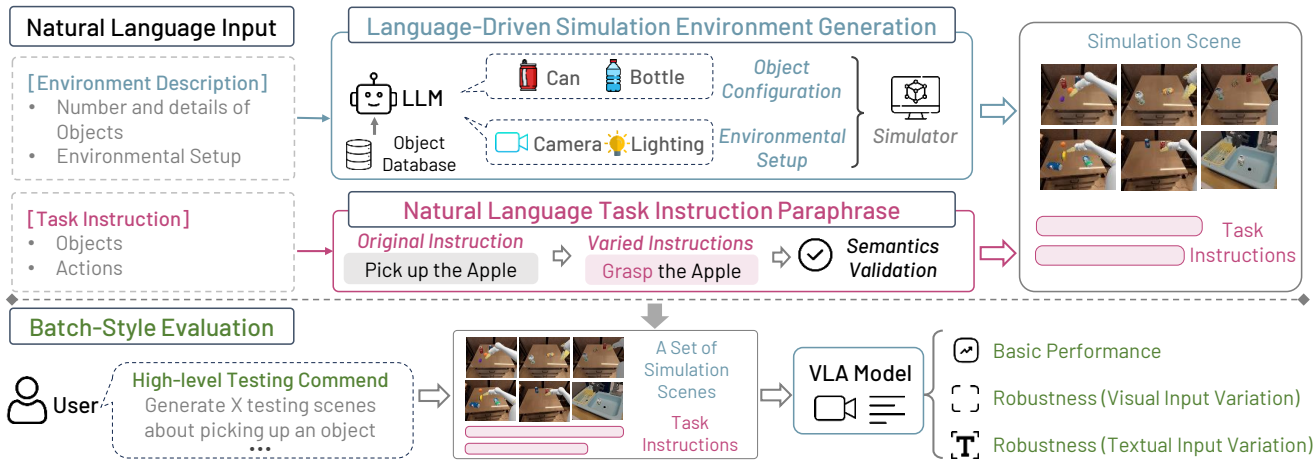

Fig. 2: Overview of LADEV. LADEV proposes: (1) Language-driven simulation environment generation; (2) Natural language task instruction paraphrase; (3) Batch-style evaluation. For details about the prompts used in this work, please refer to the preprint version or the supplementary website of this paper.

simulation environments in SimplerEnv requires manual adjustments, which can be labor-intensive when testing numerous different environments for comprehensive evaluations. Moreover, SimplerEnv alters only the simulated manipulation scenes, e.g., the objects in the environment, which solely affect the visual input to the VLA models. The natural language task instruction, i.e., a crucial input component that specifies the manipulation tasks, remains unchanged in SimplerEnv. To thoroughly test and evaluate VLA models, it is essential not only to efficiently generate a wide range of manipulation scenarios but also to create diverse natural language task instructions. These instructions should describe different tasks or express the same task using varying sentence structures and vocabulary to effectively test the language input aspect of the VLA models.

To achieve this, we propose in this work a comprehensive language-driven testing and evaluation platform called **LADEV**, that is specifically designed for VLA models. Building on SimplerEnv, we introduce three major advancements in LADEV: (1) *Language-driven Simulation Environment Generation*: instead of manual adjustments, we introduce an automated mechanism to generate simulation environments based on simple language descriptions of the desired manipulation scenarios. Using LLMs, these descriptions are translated into environmental configurations compatible with the simulator for constructing the simulation environment (see Fig. 2). To further expand simulation diversity and incorporate a wide variety of objects, we also integrate LADEV with the YCB object dataset [26], enabling the automatic selection and inclusion of appropriate object models in the simulation based on the given language input. (2) *Natural Language Task Instruction Paraphrase*: in addition to generating simulation environments, we also propose a method for paraphrasing natural language task instructions. Given an original instruction for the desired manipulation task, we use LLMs to create alternative sentences that convey the same task but with different sentence structures and wording (see Fig. 2). (3) *Batch-Style Evaluation*: to further streamline the testing and evaluation process, we implement a batch-style generation mechanism capable of creating numerous distinct test environments from a single command input. Specifically, we ask an LLM to generate a complete testing script with descriptions of diverse manipulation scenes, which are then passed to the scene generation process to assess the VLA model's performance in each individual scenario.

The contributions of this paper are summarized as follows:

- We propose a novel language-driven approach that autonomously generates simulation environments from natural language descriptions of the desired manipulation tasks. This fully automated process greatly improves the efficiency of testing and evaluating VLA models, providing a solid foundation for comprehensive performance assessments.
- We present a paraphrase mechanism that transforms the given natural language task instruction into various forms, enabling a comprehensive assessment of VLA models' ability to handle diverse language inputs. This capability fills a gap in prior evaluations of VLA models, which focused exclusively on simulation environments while neglecting the essential role of language input.
- We introduce a batch-style generation approach that is able to construct a diverse range of manipulation scenarios from a single input command. This "one-line" testing command enables rigorous large-scale testing and evaluation of VLA models in an efficient way.
- Using the proposed LADEV platform, we conduct a thorough and extensive evaluation of multiple state-of-the-art VLA models. Specifically, we examine the performance of seven VLA models on four robotic manipulation tasks using over xxxxx distinct scenes, showcasing their actual capabilities in different scenarios. (ZZ: a large about how many different scenes we have tests)

II. Related Work

A. LLM and VLM in Robotics

In recent research, LLMs have been applied to various robotic tasks, such as decision-making [6], [8], [10] and reasoning [7], [9], [27]. For instance, [6] leverages LLMs' semantic capabilities to process natural language instructions,

enabling robots to perform tasks assigned by humans through a value function. Similarly, [9] utilizes LLMs to evaluate the feasibility of task plans in a dialogue-based format, allowing robots to correct their actions as needed. Other work has explored using LLMs for task and motion planning [3], [28]–[32]. For example, [28] uses LLMs to guide object rearrangement, improving both autonomy and efficiency. Meanwhile, [5] explores the potential of LLMs with a self-refinement mechanism for long-horizon sequential task planning, increasing task success rates compared to a zero-shot LLM approach. The incorporation of LLMs significantly advances robotic intelligence, enhancing both autonomy and interaction with human users.

Extended from LLMs, an increasing number of studies now have utilized VLMs to equip robotic systems with the ability to process visual inputs [13], [33]. One common application of VLMs in robotics is reasoning about the environment and identifying interactable objects [34]–[38]. For instance, [35] combines VLM and LLM to generate 3D affordance and constraint maps that guide robotic manipulation tasks. Similarly, [36] proposes a physically grounded VLM to improve the interaction between the robot and the object. [37] introduces a VLM-based navigation approach for determining the robot's motion in human-centred environments. The integration of visual processing capabilities further enhances robots' understanding of tasks and environments, opening up the potential for achieving general robotic intelligence [17].

B. VLA Models in Robotics

VLA models are end-to-end multi-modality foundation models evolved from VLMs [17], [39], [40]. Currently, most VLA models are designed for robotic manipulation tasks, such as pick-and-place and grasping [16], [18], [20], [41]–[43]. One of the pioneering works in VLA models is RT-1 [18], which combines a FiLM EfficientNet and a transformer to learn control policies from 130k real-world robot demonstrations. RT-2 [41] advances RT-1 by introducing co-fine-tuning, integrating low-level control policies with high-level task planners to create a more comprehensive robotic system. Since the release of the Open X-Embodiment dataset [16], a series of VLA models have been developed by either training or fine-tuning on this dataset, such as OpenVLA [20], Octo [42], and LLaRA [43]. These models have demonstrated strong performance in their respective training environments, showing great potential for enabling intelligent robotic manipulation using only image and language inputs.

However, ensuring the reliability and robustness of VLA models is challenging, as their performance heavily relies on the quality of the training data [2]. This necessitates an extensive testing and evaluation platform specifically designed for VLA models. As previously mentioned, the SimplerEnv, introduced in [23], provides valuable simulation environments. However, it requires manual adjustments for environment construction and neglects the impact of language inputs. To overcome these limitations, we therefore propose LADEV in this work, which enables a more efficient, comprehensive, and automated evaluation process for VLA models.

III. LADEV

In this section, we explain details about the proposed LADEV platform. First, we describe how LLMs are used to generate simulation environments from natural language descriptions of the desired manipulation scenarios. Next, we introduce a paraphrase mechanism that alters the given natural language task instructions. Lastly, we present a batch-style evaluation method that greatly accelerates the evaluation process with improved efficiency. Due to page limit, detailed information about the prompts used in this work is presented in the preprint version and the website[1] of this paper.

A. Language-Driven Simulation Environment Generation

The core concept behind the automated generation of simulated manipulation environments is to convert natural language descriptions into simulator-compatible environmental configurations by leveraging LLMs. To achieve this, we use a fixed structure for the natural language description, which includes the following components (see also Fig. 3):

- *Number and details of objects*: First, we specify the total number of objects and provide additional details, such as their specific types and poses, to be included in the simulation environment. If object details are not needed, this part can be left blank.
- *Environmental setup*: Then, we describe the environmental setup, including the lighting condition and camera pose. If not specified, predefined default values will be used.

Using this structured description, we apply a two-step process to separately handle the object configuration and environmental setup during the generation process.

1) Object Configuration: We begin by using the descriptions of the number and details of objects to select appropriate models from LADEV's object model database, which combines the YCB object dataset [26] and the default dataset from SimplerEnv [23]. This is achieved by providing a predefined list of all available objects, along with the natural language description, to the LLM, which then generates a list of object addition operations. The length of this list corresponds to the specified number of objects, and each entry represents the addition of an object model to the simulation environment (see Fig. 3). If detailed object specifications are provided, the LLM prioritizes selecting models that best match the criteria. For example, if the user requests a *coke can*, LADEV searches for the relevant model and adds it if available. When no specific details are given, random objects are selected. Similarly, if specific object poses are provided, they are translated into corresponding coordinates; otherwise, random values within a predefined range are assigned.

2) Environmental Setup: We then prompt the LLM with the description of the environmental setup to configure the simulation parameters. In the current version of LADEV, two environmental configurations are considered: the lighting condition and the camera pose. If a specific value is provided for the lighting condition, the LLM generates an operation command to adjust the scene's lighting intensity accordingly.

[1] https://sites.google.com/view/ladev

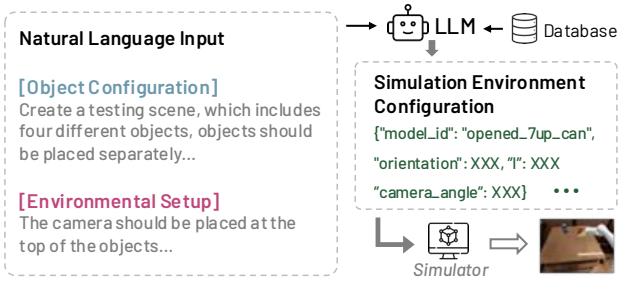

Fig. 3: Example of language-driven simulation environment generation.

Similarly, if a camera pose is specified, the LLM generates an operation to move or rotate the camera to match the desired pose, ensuring proper visual inputs for the VLA models. If no information is given for these parameters, predefined default values are applied.

Once the object addition and environmental adjustment operations are generated, we pass them to the simulator to construct the corresponding simulation environment. To enhance the accuracy of the LLM's translations, we employ few-shot in-context learning [44] in our prompts. This approach also ensures that the LLM outputs are formatted in a way that is compatible with the simulator. In LADEV, these operations are provided in JSON configuration formats. This language-driven generation process reduces the need to manually adjust the underlying code used to construct simulation environments, significantly saving time and effort. Additionally, it facilitates efficient evaluation of VLA models across a variety of manipulation scenarios, which we later propose to execute in a batch-style process.

B. Natural Language Task Instruction Paraphrase

To evaluate the performance of VLA models in processing diverse language inputs, we also propose a paraphrase mechanism that generates varied natural language task instructions. The paraphrase mechanism consists of two phases: a generation phase and a validation phase (see Fig. 2).

The input to the generation phase is an original task instruction that follows the standard format used in previous works [18], [20], [41], [42], such as using *"pick up apple"* to describe a task involving picking up an apple. The goal of the generation phase is to produce a predefined number, k, of alternative instructions that convey the same meaning but differ in sentence structure and wording. For example, the original instruction *"pick up apple"* could be rephrased as *"grasp apple"*, *"let's pick the apple"*, or *"can you lift the apple"*, etc. This is achieved by prompting an LLM with the original instruction and guidelines for generating alternative sentences. The LLM then outputs k variations of the task instructions with distinct wordings and structures.

After generating a set of candidate sentences, we implement a validation phase to ensure that each sentence accurately retains the same meaning as the original, minimizing potential errors. This is achieved by using a *sentence BERT* model [45] for similarity checking. Specifically, the *sentence BERT* model calculates the semantic similarity between the original task instruction and the candidate variations. If the similarity value exceeds a predefined threshold, the varied task instruction is considered to have the same meaning and is deemed valid. These valid instructions are then used to evaluate the VLA models' performance in handling diverse language inputs.

By utilizing the proposed natural language task instruction paraphrase mechanism, we significantly increase the diversity of language inputs for VLA models. This approach addresses a crucial gap in the overall evaluation, focusing on the previously overlooked aspect of assessing how natural language instructions impact the performance of VLA models.

C. Batch-Style Evaluation

Through the methods proposed in Sec.III-A and Sec.III-B, we can generate a single manipulation scenario with multiple diverse language task instructions that convey the same task. However, for a comprehensive evaluation of the VLA model, it is crucial to test its performance across various scenarios and tasks. To expedite this process, we introduce a batch-style evaluation mechanism that automatically generates a specified number of distinct manipulation scenarios from a single natural language input.

Specifically, we instruct the LLM to automatically generate diverse language inputs for executing both the simulation environment generation and task instruction paraphrase processes (see Fig. 2). Suppose the user wishes to create n test scenes with k task instructions per scene. In this case, the LLM is prompted to randomly generate n sets of natural language inputs that specify objects and environmental setups, along with n original task instructions following the standard format. Each original task instruction is then used in the paraphrase process to generate k variations, resulting in a total of $n \times k$ language inputs. Due to page limitations, the detailed prompt templates and examples used for in-context learning are provided on the website accompanying this paper.

The proposed batch-style evaluation mechanism enables efficient large-scale testing of VLA models, facilitating a more comprehensive and reliable assessment of their robustness and effectiveness. In the next section, we apply this mechanism to evaluate the performance of multiple state-of-the-art VLA models across various manipulation scenarios.

IV. Experiments

In this section, we first present the details of our experimental setup. Then, we utilize a concrete example to illustrate how LADEV generates a simulation environment from a natural language input and paraphrases the task instruction. Finally, we conduct large-scale experiments to assess the performance of multiple state-of-the-art VLA models. This demonstrates the efficiency and effectiveness of the proposed LADEV for evaluating VLA models.

A. Experimental Setup

We consider four robotic manipulation tasks in LADEV for evaluating VLA models: (1) *Pick up an object*; (2) *Move object A near object B*; (3) *Put object A on object B*; and (4) *Put object A inside object B*. For performance evaluation,

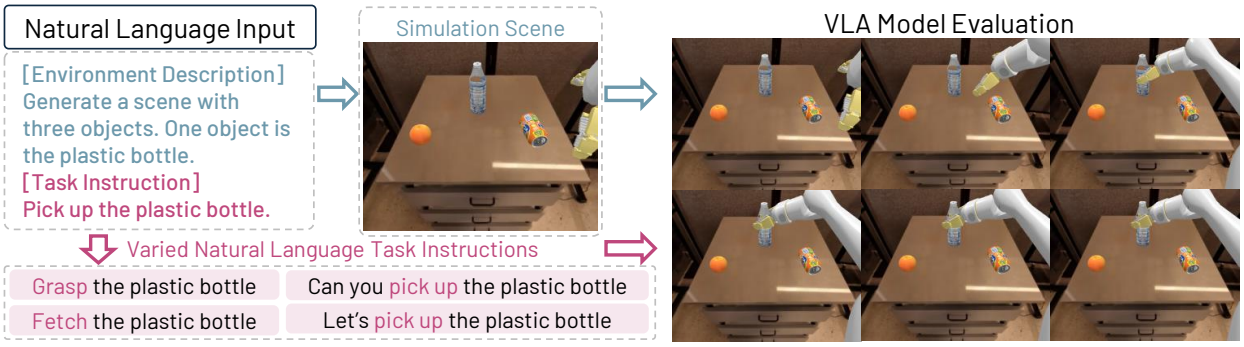

Fig. 4: Example of simulation scene generation and natural language instruction paraphrase with LADEV. See also the supplementary video for more details.

we assess the impact of the following factors on the success rates of completing the specified manipulation tasks:

- *Number of objects*: We first investigate how the number of objects in the simulation scene affects performance. More objects introduce additional obstacles, increasing the difficulty for VLA models in correctly identifying the target object. To evaluate this, we generate 100 test scenes for each task, with each scene containing 1 to 5 objects. Only basic task instructions are used to focus on the effect of the number of objects.
- *Task instructions*: For each task, we generate 100 test scenes with 1 to 4 randomly selected objects. Each scene is evaluated using both the basic task instructions and those paraphrased by LADEV to examine the influence of varied language inputs on VLA models' performance. (ZZ: how many varied?)
- *Unseen objects*: The LADEV's object model database is the combination of the SimplerEnv [23] (18 objects) and the YCB [46] (65 objects). Objects in SimplerEnv are generally considered part of the VLA models' training dataset, while objects from the YCB can be regarded as unseen. We evaluate the effect of unseen objects by generating two groups of 100 test scenes for each task, randomly including 1 to 4 objects from either the SimplerEnv or the YCB.
- *Environmental conditions*: Lastly, we assess the impact of environmental conditions. For each task, we create three sets of 100 test scenes with 1 to 4 objects: one with default lighting and camera settings, one with randomly adjusted lighting conditions, and one with altered camera poses. To ensure all objects remain visible and recognizable, only small adjustments are made to lighting conditions and camera poses. (ZZ: how small?)

We compare the following state-of-the-art VLA models: RT-1-1k, RT-1-58k, RT-1-400k [18], RT-1-X [16], Octo-base, Octo-small [42], and OpenVLA-7b [20]. We use GPT-4o as our LLM. All experiments are conducted on a server with an AMD 5955WX CPU and two NVIDIA RTX A6000 GPUs. For more details about the experimental setup, please refer to the supplementary website of this paper.

B. Environment Generation and Command Mutation

We first use a concrete example to demonstrate the processes of simulation environment generation and natural language task instruction paraphrase. As shown in Fig. 4, the natural language input instructs LADEV to create a simulation environment with three objects, one of which is a plastic bottle. After searching the model database, LADEV generates the required environment. The basic task instruction, *"pick up the plastic bottle"*, is also paraphrased into four variations by LADEV. These task instructions, along with the generated environment, are used to evaluate the performance of VLA models. This individual process can also be scaled using the proposed batch-style evaluation mechanism, which automatically generates multiple manipulation scenes, each with varied task instructions, to comprehensively test the VLA models' performance.

C. Performance Evaluation of VLA Models

Using the proposed LADEV, we perform a large-scale evaluation of VLA models by considering the aforementioned factors. The success rates for completing the given tasks under various conditions are shown in Table I-Table IV. We highlight the following key observations:

- *Number of objects*: As shown in Table I, all VLA models perform best when only one object is present, i.e., only the target object. The performance of the VLA models declines as the number of objects increases. When five objects are included, almost all models perform poorly across all test scenes and tasks. Model-wise, RT-1-58k, RT-1-400k, and RT-1-X outperform the other models on the *Pick Up* task, while OpenVLA-7b achieves the highest performance on the *Move Near* task. However, for the *Put On* and *Put In* tasks, all VLA models show poor results, with success rates below 5%.
- *Task instructions*: From Table II, we observe a significant performance drop when using paraphrased instructions compared to the basic ones in the *Pick Up* and *Move Near* tasks. However, the performance differences in the *Put On* and *Put In* tasks are minimal, as the basic instructions already result in poor performance.
- *Unseen objects*: As shown in Table III, VLA models perform worse with YCB objects compared to SimplerEnv objects. For instance, models like RT-1-58k, RT-1-400k, RT-1-X, and OpenVLA-7b perform relatively well in the *Pick Up* and *Move Near* tasks with SimplerEnv objects. However, when manipulating YCB objects, their performance drops by 10% to 30%.
- *Environmental conditions*: Table IV demonstrates that even small changes in environmental conditions can significantly

TABLE I: VLA models' performance with different numbers of objects.

VLA Model	Pick Up						Move Near						Put On						Put In					
	1	2	3	4	5	Avg.	1	2	3	4	5	Avg.	1	2	3	4	5	Avg.	1	2	3	4	5	Avg.
RT-1-1k	0%	1%	0%	0%	0%	0.2%	2%	2%	1%	1%	1%	1.4%	0%	0%	0%	0%	0%	0.0%	0%	0%	0%	0%	0%	0.0%
RT-1-58k	36%	41%	27%	23%	21%	29.6%	11%	11%	9%	10%	6%	9.4%	0%	0%	0%	0%	0%	0.0%	0%	0%	0%	0%	0%	0.0%
RT-1-400k	44%	37%	35%	33%	26%	35.0%	12%	14%	4%	5%	3%	7.6%	0%	0%	0%	0%	0%	0.0%	0%	0%	0%	1%	0%	0.2%
RT-1-X	26%	30%	19%	16%	9%	20.0%	7%	12%	4%	2%	4%	5.8%	3%	1%	1%	1%	2%	1.6%	0%	0%	0%	1%	0%	0.2%
Octo-small	2%	0%	1%	0%	0%	0.6%	3%	1%	6%	0%	0%	2.0%	4%	5%	6%	1%	2%	3.6%	0%	3%	0%	0%	2%	1.0%
Octo-base	1%	0%	0%	0%	0%	0.2%	0%	0%	0%	1%	0%	0.2%	0%	0%	5%	0%	1%	1.2%	0%	1%	3%	0%	1%	1.0%
OpenVLA-7b	12%	7%	8%	7%	2%	7.2%	23%	18%	12%	8%	2%	12.6%	1%	5%	1%	1%	2%	2.0%	5%	1%	1%	4%	0%	2.2%

TABLE II: Basic task instructions vs. paraphrased (Para.) task instructions.

VLA Model	Pick Up		Move Near		Put On		Put In	
	Basic	Para.	Basic	Para.	Basic	Para.	Basic	Para.
RT-1-1k	0%	2%	3%	1%	0%	0%	0%	0%
RT-1-58k	28%	17%	12%	6%	0%	1%	1%	0%
RT-1-400k	36%	22%	7%	3%	0%	1%	0%	0%
RT-1-X	20%	13%	7%	4%	2%	0%	1%	0%
Octo-base	0%	0%	2%	0%	2%	3%	3%	1%
Octo-small	0%	1%	2%	5%	4%	3%	1%	1%
OpenVLA-7b	8%	7%	12%	4%	2%	4%	1%	2%

TABLE III: Objects from SimplerEnv (SE) vs. objects from YCB

VLA Model	Pick Up		Move Near		Put On		Put In	
	SE	YCB	SE	YCB	SE	YCB	SE	YCB
RT-1-1k	0%	0%	3%	0%	0%	0%	0%	1%
RT-1-58k	28%	2%	12%	7%	0%	0%	1%	0%
RT-1-400k	36%	5%	7%	3%	0%	2%	0%	0%
RT-1-X	20%	3%	7%	0%	2%	2%	1%	1%
Octo-base	0%	0%	2%	0%	2%	1%	3%	0%
Octo-small	0%	0%	2%	0%	4%	0%	1%	3%
OpenVLA-7b	8%	0%	12%	6%	2%	0%	1%	0%

impact model performance. For lighting condition changes, the RT-1 and RT-1-X models are more adversely affected compared to other models. However, for camera pose adjustments, no notable performance differences are observed.

V. Discussion

A. Pros and Cons of VLA Models

By combining visual, linguistic, and action-based information, VLA models offer several advantages to robotic systems. For instance, they enable robots to better interpret their surroundings and execute tasks using natural language instructions, minimizing the reliance on hardcoded or structured inputs. This facilitates more intuitive human-robot communication, enhancing interaction flexibility as well as the robots' autonomy and intelligence. However, VLA models also face multiple challenges. For example, training these models requires large, multi-modal datasets and significant computational resources. Unfortunately, high-quality datasets that align visual inputs, language descriptions, and corresponding actions are scarce. Moreover, our experiments show that current VLA models still struggle with even simple manipulation tasks under varied conditions, highlighting the need for deeper exploration in this area to achieve better performance.

B. Limitations and Future Work

One limitation of the current version of LADEV is that it only considers four manipulation tasks. This is primarily due to the fact that state-of-the-art VLA models are still only trained for simple tasks. Another drawback is that our evaluations are conducted solely in simulations, as performing comprehensive real-world experiments is difficult and resource-intensive. To reduce the gap between simulation and reality, we adopt the same approach as SimplerEnv by using real-world images as backgrounds for the visual inputs to the VLA models. However, further research is needed to better minimize the simulation-to-reality gap and develop more efficient methods for assessing the performance of VLA models in real-world conditions.

TABLE IV: Influence of different environmental conditions: default (De.), mutated lighting (Li.), and mutated camera poses (Ca.) ZW: to complete the results

VLA Model	Pick Up			Move Near			Put On			Put In		
	De.	Li.	Ca.	De.	Li.	Ca.	De.	Li.	Ca.	De.	Li.	Ca.
RT-1-1k	0%	0%	1%	3%	3%	6%	0%	0%	2%	0%	0%	
RT-1-58k	28%	23%	31%	12%	12%	11%	0%	0%	0%	1%	1%	
RT-1-400k	36%	20%	30%	7%	6%	8%	0%	0%	0%	0%	0%	
RT-1-X	20%	9%	14%	7%	8%	7%	2%	2%	1%	1%	1%	
Octo-base	0%	0%	1%	2%	2%	3%	2%	2%	3%	3%	3%	
Octo-small	0%	1%	0%	2%	2%	2%	4%	4%	2%	1%	1%	
OpenVLA-7b	8%	12%	14%	12%	12%	15%	2%	2%	1%	1%	1%	

For future work, we plan to expand the LADEV platform by incorporating more object models and manipulation scenarios, which would greatly increase its diversity and effectiveness. Another potential direction is to compare the performance of VLA models in both simulation and real-world environments across several representative tasks, serving as an indicator of their reliability in practical applications. However, how to select the most appropriate and representative tasks will require further in-depth research.

VI. Conclusion

In this work, we propose LADEV, a testing and evaluation platform for VLA models in robotic manipulation tasks. By introducing a language-driven mechanism, we efficiently generate simulation environments from simple natural language inputs, mitigating the need for manual adjustments. To assess the impact of language instructions on VLA models, we also present a task instruction paraphrase approach that automatically generates diverse sentences to enrich the language input. Moreover, to further improve the evaluation efficiency, we develop a batch-style evaluation mechanism that creates multiple testing scenarios from a single command, enabling a comprehensive and streamlined assessment of VLA models' performance. Our platform notably improves the evaluation process and establishes a strong baseline for advancing VLA models, paving the way for more intelligent robotic systems with enhanced autonomy and decision-making capabilities.

REFERENCES

[1] F. Zeng, W. Gan, Y. Wang, N. Liu, and P. S. Yu, "Large language models for robotics: A survey," *arXiv preprint arXiv:2311.07226*, 2023.

[2] J. Wang, Z. Wu, Y. Li, H. Jiang, P. Shu, E. Shi, H. Hu, C. Ma, Y. Liu, X. Wang *et al.*, "Large language models for robotics: Opportunities, challenges, and perspectives," *arXiv preprint arXiv:2401.04334*, 2024.

[3] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*. PMLR, 2022.

[4] S. S. Kannan, V. L. Venkatesh, and B.-C. Min, "Smart-llm: Smart multi-agent robot task planning using large language models," *arXiv preprint arXiv:2309.10062*, 2023.

[5] Z. Zhou, J. Song, K. Yao, Z. Shu, and L. Ma, "Isr-llm: Iterative self-refined large language model for long-horizon sequential task planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2081–2088.

[6] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[7] J. Song, Z. Zhou, J. Liu, C. Fang, Z. Shu, and L. Ma, "Self-refined large language model as automated reward function designer for deep reinforcement learning in robotics," *arXiv preprint arXiv:2309.06687*, 2023.

[8] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: From natural language instructions to feasible plans," *arXiv preprint arXiv:2303.12153*, 2023.

[9] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[10] S. Li, X. Puig, C. Paxton, Y. Du, C. Wang, L. Fan, T. Chen, D.-A. Huang, E. Akyürek, A. Anandkumar *et al.*, "Pre-trained language models for interactive decision-making," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 199–31 212, 2022.

[11] W. Street, "Llm theory of mind and alignment: Opportunities and risks," *arXiv preprint arXiv:2405.08154*, 2024.

[12] Y. Du, Z. Liu, J. Li, and W. X. Zhao, "A survey of vision-language pre-trained models," *arXiv preprint arXiv:2202.10936*, 2022.

[13] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[14] C. Cui, Y. Ma, X. Cao, W. Ye, Y. Zhou, K. Liang, J. Chen, J. Lu, Z. Yang, K.-D. Liao *et al.*, "A survey on multimodal large language models for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 958–979.

[15] Z. Long, G. Killick, R. McCreadie, and G. Aragon-Camarasa, "Robollm: Robotic vision tasks grounded on multimodal large language models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 428–12 435.

[16] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[17] Y. Ma, Z. Song, Y. Zhuang, J. Hao, and I. King, "A survey on vision-language-action models for embodied ai," *arXiv preprint arXiv:2405.14093*, 2024.

[18] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[19] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[20] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[21] Z. Xu, K. Wu, J. Wen, J. Li, N. Liu, Z. Che, and J. Tang, "A survey on robotics with foundation models: toward embodied ai," *arXiv preprint arXiv:2402.02385*, 2024.

[22] D. Myers, R. Mohawesh, V. I. Chellaboina, A. L. Sathvik, P. Venkatesh, Y.-H. Ho, H. Henshaw, M. Alhawawreh, D. Berdik, and Y. Jararweh, "Foundation and large language models: fundamentals, challenges, opportunities, and social impacts," *Cluster Computing*, vol. 27, no. 1, pp. 1–26, 2024.

[23] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani *et al.*, "Evaluating real-world robot manipulation policies in simulation," *arXiv preprint arXiv:2405.05941*, 2024.

[24] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "SAPIEN: A simulated part-based interactive environment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[25] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," *arXiv preprint arXiv:2302.04659*, 2023.

[26] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.

[27] A. Zeng, M. Attarian, B. Ichter, K. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani *et al.*, "Socratic models: Composing zero-shot multimodal reasoning with language," *arXiv preprint arXiv:2204.00598*, 2022.

[28] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," *arXiv preprint arXiv:2303.06247*, 2023.

[29] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *International Conference on Robotics and Automation*. IEEE, 2023, pp. 11 523–11 530.

[30] T. Silver, S. Dan, K. Srinivas, J. B. Tenenbaum, L. P. Kaelbling, and M. Katz, "Generalized planning in pddl domains with pretrained large language models," *arXiv preprint arXiv:2305.11014*, 2023.

[31] T. Silver, V. Hariprasad, R. S. Shuttleworth, N. Kumar, T. Lozano-Pérez, and L. P. Kaelbling, "Pddl planning with pretrained large language models," in *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[32] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "Llm+ p: Empowering large language models with optimal planning proficiency," *arXiv preprint arXiv:2304.11477*, 2023.

[33] Y. Du, K. Konyushkova, M. Denil, A. Raju, J. Landon, F. Hill, N. de Freitas, and S. Cabi, "Vision-language models as success detectors," *arXiv preprint arXiv:2303.07280*, 2023.

[34] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[35] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[36] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[37] D. Song, J. Liang, A. Payandeh, X. Xiao, and D. Manocha, "Vlm-social-nav: Socially aware robot navigation through scoring using vision-language models," *arXiv preprint arXiv:2404.00210*, 2024.

[38] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "Ok-robot: What really matters in integrating open-knowledge models for robotics," *arXiv preprint arXiv:2401.12202*, 2024.

[39] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu *et al.*, "Vision-language foundation models as effective robot imitators," *arXiv preprint arXiv:2311.01378*, 2023.

[40] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia *et al.*, "Open-world object manipulation using pre-trained vision-language models," *arXiv preprint arXiv:2303.00905*, 2023.

[41] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[42] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[43] X. Li, C. Mata, J. Park, K. Kahatapitiya, Y. S. Jang, J. Shang, K. Ranasinghe, R. Burgert, M. Cai, Y. J. Lee *et al.*, "Llara: Supercharging robot learning data for vision-language policy," *arXiv preprint arXiv:2406.20095*, 2024.

[44] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[45] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[46] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.