A Preliminary Evaluation of the Non-determinism of VLAs for Robotic Tasks

Antonin Rousseau*, Deyun Lyu†, Paolo Arcaini†, Fuyuki Ishikawa†, Gilles Perrouin*
*University of Namur, Namur, Belgium
†National Institute of Informatics, Tokyo, Japan
Email: antonin.rousseau@student.unamur.be, gilles.perrouin@unamur.be, {lyudeyun, arcaini, f-ishikawa}@nii.ac.jp

***Abstract*—Vision-Language-Action (VLA) models combine vision capabilities with language understanding to guide robots in a closed-loop to accomplish a given high-level task. Succession of VLAs stochastic inference steps may propagate deviations in generated actions within and across runs, leading to non-determinism. This paper investigates non-determinism for two VLAs interacting with a robotic arm in a simulated environment. We introduce specific metrics and find that VLAs exhibit contrasted non-deterministic behaviour both for task outcomes and position of target objects throughout simulation steps. This study calls for systematic study of non-determinism in VLA testing.**

***Index Terms*—Vision-language-action models, robotic manipulation, non-determinism, simulation-based testing**

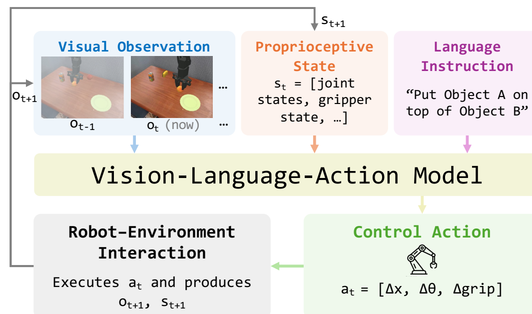


Fig. 1: Closed-loop execution of a VLA

I. Introduction

Vision-Language-Action models (VLAs) integrate visual perception, language understanding, and action generation within a unified framework, and are increasingly applied to real-world robotic manipulation tasks [6]. Given visual observations and language instructions, they generate actions that enable robots to interact with their environments and complete manipulation tasks. Because these models directly control robotic systems, failures may cause unsafe motions, leading to task failure, object damage, or even harm to nearby people. Thorough testing is therefore important before real-world deployment.

Unlike single-step prediction tasks, robotic manipulation powered by VLAs involves successive inferences and sequential interaction with the environment. At each step, the model outputs a control action based on the current observation, and the executed action evolves the robot state and determines the observation for the next inference step. Fig. 1 illustrates this closed-loop interaction.

Even under identical test conditions, repeated executions of a VLA may produce slightly different actions. Such differences may originate from stochastic action generation or non-deterministic numerical computation during model inference. As actions affect subsequent robot states and observations in closed-loop execution, initially minor differences may propagate and accumulate over time, leading to divergent action sequences, trajectories, interactions, or final task outcomes. This potential for behavioural divergence motivates an assessment of the *non-determinism* of VLAs in robotic manipulation.

P. Arcaini is supported by the ASPIRE grant No. JPMJAP2301, JST. Gilles Perrouin is an FNRS Research Associate.

Recent studies have explored testing VLAs for robotic manipulation. Wang et al. [9] propose *VLATest*, which generates test scenarios by varying manipulated objects, distractor objects, lighting conditions, and camera settings. Valle et al. [7] propose a set of fine-grained metrics for evaluating execution quality and uncertainty. Valle et al. [8] also propose a metamorphic testing approach for VLAs, i.e., modelling expected consequences of changing inputs as metamorphic relationships and ensuring those relationships are not violated. Despite these advances, those studies evaluate each test configuration based on a single execution and do not consider the potential non-determinism across repeated executions. Such an evaluation implicitly assumes that a single execution is representative of the model's behaviour under that configuration. However, when repeated executions exhibit different behaviours, the reported result may depend on the particular run being observed. A single execution may therefore fail to reveal the range of possible behaviours, miss occasional failures, and provide an unstable estimate of model performance. In particular, the same test may succeed in some runs but fail in others. Therefore, single-run evaluation provides a limited characterization of the behavioural consistency and reliability of VLAs.

To address this gap, ***we conduct a systematic empirical study of the non-deterministic behaviour of VLAs in robotic manipulation tasks***. We evaluate two representative VLAs in a deterministic simulation environment using a set of 750 test configurations, each executed three times. In each configuration, a robotic arm should move an item (a can) onto a plate in the presence of varying numbers of *distractor objects*, thereby reflecting more realistic industrial environments [5]. We study non-determinism both at the outcome level (success rate) and during execution (object trajectories). Specifically, we investigate (i) how inconsistent success and failure outcomes affect the final item placement, (ii) when trajectory divergence emerges, and (iii) how it evolves throughout execution. We offer four metrics to characterize outcome and trajectory non-determinism and find that the two tested VLAs exhibit different levels of non-determinism: `GR00T N1` [1] is more non-deterministic and sensitive to distractor objects than `SpatialVLA` [4] across executions.

II. Study Design

Regarding outcome non-determinism, we report whether the test has succeeded or failed and the time necessary for accomplishing the task, measured in simulation steps. The consistency of those metrics will be evaluated by repeating the exact same test with the same model several times.

Regarding trajectory non-determinism, we observe the location of the target object throughout the test across repetitions. Our motivation is to identify when and how divergence happens on a same test, how it evolves over time and how the outcome is changed.

Code and all results are available at our GitHub repository https://github.com/rouant/determinism_of_vlas.

A. Research Questions (RQs)

RQ1: *To what extent do VLAs exhibit non-deterministic behaviour across repeated executions?*

RQ1.a: *What is the overall degree of non-determinism across all test configurations?*

RQ1.b: *How does non-determinism vary across test suites with different levels of difficulty?*

RQ1.c: *How does non-determinism vary with simulation length?*

In this RQ, we investigate the overall degree of non-determinism and analyze how it varies with test difficulty and simulation length.

RQ2: *When does trajectory divergence first emerge during execution, and how does it evolve over time?*

In this RQ, we compare robot trajectories across repeated executions to identify when divergence first occurs and how it develops throughout the execution.

B. Subject VLA Models

We use two representative VLAs for the study. `GR00T N1` [1] follows a dual-architecture design: a high-level planner, built on a VLM backbone and operating at a lower frequency, interprets the task and the environment, while a low-level policy, operating at a higher frequency, produces the corresponding actions. We select `GR00T N1` as its original evaluation reports strong generalization capabilities and superior performance over other VLAs. `SpatialVLA` [4] achieves superior performances through enhanced 3D spatial understanding, allowing it to integrate spatial context such as "the closest item". Except for its *Ego3D Position Encoding*, `SpatialVLA` follows the common VLA architecture, best exemplified by OpenVLA [2], built upon a VLM backbone.

C. Robotic Manipulation Task and Simulation Environment

The task we consider is **"Place Object A on top of Object B"**, which requires grasping a designated object, referred to as the *target object*, and placing it on top of a *receptacle*, such as a plate. We choose this task for its composite nature, as it requires the policy to compose several sub-skills: identifying the correct target and receptacle, grasping the target object, transporting it, and placing it accurately on the receptacle.

To run VLAs, we use a SIMPLER environment [3] tailored for our task. SIMPLER provides environments that mitigate the *real-to-sim gap*, i.e., the difference between simulation and the real world. We use the fork of the project from [7].

D. Test Suite Construction

To generate a test, we randomly sample a valid scene configuration from the space of valid configurations $\mathcal{T}$. Each object is represented as a pair $\langle \mathrm{c}, \mathrm{p}\rangle$, where c denotes its category and $\mathrm{p} = (\mathrm{x}, \mathrm{y})$ denotes its position in the scene. A test is defined as $t = (\mathrm{obj}_{to}, \mathrm{obj}_{ro}, \mathrm{O}_{do}) \in \mathcal{T}$, where:

$\mathrm{obj}_{to} = \langle \mathrm{c}_{to}, \mathrm{p}^0_{to}\rangle$ is the target object, where p^0_{to} denotes its initial position; the target object is moved during execution and its position is tracked over time (see Sect. II-E);

$\mathrm{obj}_{ro} = \langle \mathrm{c}_{ro}, \mathrm{p}_{ro}\rangle$ is the receptacle on which the target object should be placed;

$\mathrm{O}_{do} = \{\langle \mathrm{c}_{do,i}, \mathrm{p}_{do,i}\rangle\}_{i=1}^{N_{do}}$, with $N_{do} \in \{0, \ldots, 4\}$, is the set of distractor objects. When $N_{do} = 0$, the scene contains only the target object and the receptacle. Distractor objects are objects that are irrelevant to the task and should not be interacted with, e.g., the robotic arm should not touch them. They increase the scene complexity and may divert the VLA from the intended task.

The target object type and distractor-object type $\mathrm{c}_{to}, \mathrm{c}_{do,i} \in$ {*cylindric*, *cubical*, *rectangular*, *spherical*, *elongated*}, while the receptacle type $\mathrm{c}_{ro} \in$ {*round*, *square*}.

All object positions are selected from an 8×5 grid of valid locations evenly distributed across the scene, yielding 40 candidate positions. To ensure valid configurations, no two objects may overlap or occupy adjacent grid cells, including diagonally adjacent cells.

For each value of N_{do}, we randomly generate 150 valid configurations, resulting in a total of 750 tests per model.

E. Evaluation Metrics

Given a test t (see Sect. II-D), let $\mathcal{R}(t) = \{r_1, \ldots r_N\}$ denote its N repeated executions. In this study, we set $N = 3$. For each execution r_i, we record the trajectory of the target object obj_{to} as $\tau_i = [\mathrm{p}^1_{to,i}, \ldots, \mathrm{p}^{L_i}_{to,i}]$, where $\mathrm{p}^k_{to,i} = (\mathrm{x}^k_{to,i}, \mathrm{y}^k_{to,i})$ are the normalized two-dimensional coordinates of obj_{to} at simulation step k of execution r_i, and L_i is the length of the trajectory τ_i, i.e., the number of simulation steps in execution r_i. In this work, we set a fixed L_i of 120.

We evaluate the degree of non-determinism using the following four metrics, computed across repeated executions of the same test.

1) Final Position Dispersion (FPD): It measures the variation of the target object's final position across repeated executions. Given a test t, FPD is defined as:

$$FPD(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathrm{p}_{to,i}^{L_i} - \overline{\mathrm{p}_{to}^{L_i}}\right\|_2^2}, \quad \text{with } \overline{\mathrm{p}_{to}^{k}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{p}_{to,i}^{k}$$

FPD allows us to measure the difference in the final result of the execution in regard to the target object.

2) Repetition Success Rate (RSR): It measures the proportion of successful executions. Let $s_i = 1$ if execution r_i succeeds, $s_i = 0$ otherwise. Given a test t, RSR is defined as:

$$RSR(t) = \frac{1}{N}\sum_{i=1}^{N} s_i$$

Since each test is repeated three times, $RSR(t) \in \{1, 0, \frac{2}{3}, \frac{1}{3}\}$. Values of 1 and 0 indicate deterministic outcomes, as all repetitions produce the same result, whereas values of $\frac{2}{3}$ and $\frac{1}{3}$ indicate non-deterministic outcomes. For simplicity, we will identify these values as *All passing*, *None passing*, *2 passing*, and *1 passing*. We use RSR to group tests according to their success patterns across repetitions.

3) Average Length (AL): It is the average number of simulation steps across repetitions. Given a test t, AL is defined as:

$$AL(t) = \frac{1}{N}\sum_{i=1}^{N} SL(\tau_i),$$

with

$$SL(\tau_i) = \begin{cases} L_i + 1 & \text{if } r_i \text{ fails at step } L_i \\ \min\{k \mid \text{target obj. is on receptacle at } \mathrm{p}_{to,i}^{k}\} & \text{otherwise} \end{cases}$$

AL allows for quantifying the time necessary for a test to be completed by the VLA.

4) Stepwise Position Dispersion (SPD): It is similar to FPD, except that it is computed for each step instead of the final step only. Given a test t, SPD is defined as:

$$SPD(t) = \langle \sigma_1, \ldots \sigma_{L_i} \rangle \quad \text{with } \sigma_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathrm{p}_{to,i}^{k} - \overline{\mathrm{p}_{to}^{k}}\right\|_2^2}$$

We will identify with $SPD[k]$ the SPD value at time k.

F. Running of the experiments

Experiments were conducted on a server with an Intel Core i9-14900K CPU, 128 GB RAM, and an NVIDIA RTX PRO 6000 GPU. Running the experiments for `SpatialVLA` took around 150 hours, and 45 hours for `GR00T N1`.

III. Experimental results

A. Answer to RQ1

1) RQ1.a: Fig. 2 reports the FPD (see Sect. II-E1) for the two models. We observe a drastic difference between the two models: while `GR00T N1` shows a non-negligible FPD for its executions, `SpatialVLA` instead has an almost

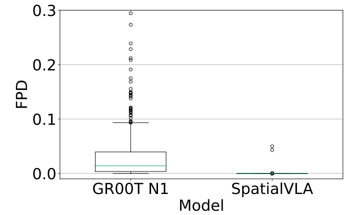
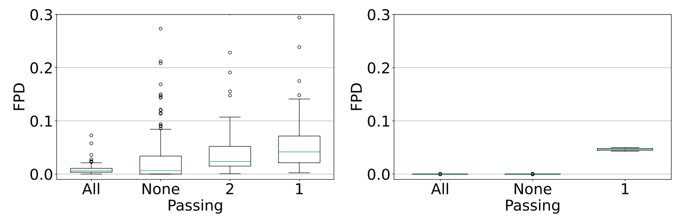


Fig. 2: RQ1.a – Distribution of FPD by model

TABLE I: RQ1.b – Distribution of tests across RSR category

# distractor objects	GR00T N1				SpatialVLA			
	Passing				Passing			
	All	None	2	1	All	None	2	1
0	35	56	28	31	30	119	0	1
1	39	53	24	34	35	115	0	0
2	38	58	28	26	39	111	0	0
3	27	60	29	34	29	120	0	1
4	38	60	28	24	45	105	0	0
Total	177	287	137	149	178	570	0	2
	464 (61.9%)		286 (38.1%)		748 (99.7%)		2 (0.3%)	



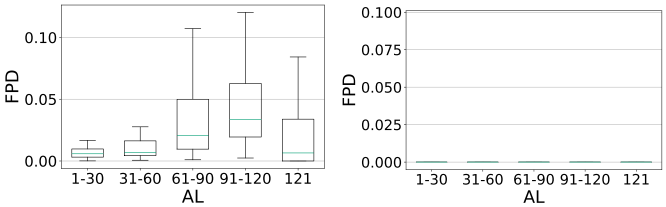
(a) `GR00T N1` (b) `SpatialVLA`

Fig. 3: RQ1.b – Distribution of FPD across RSR categories

zero FPD except for two outliers. This suggests that non-determinism greatly depends on the model and that nearly deterministic executions for VLA-based robotic manipulation are not impossible (as in `SpatialVLA`).

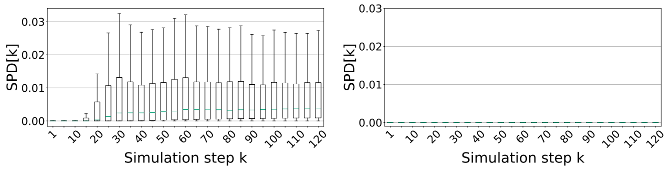
2) RQ1.b: In this RQ, we analyze the FPD of the tests by grouping them in terms of RSR (Sect. II-E2). Table I reports how tests distribute in terms of RSR. Regarding the determinism, we observe that `GR00T N1` is very non-deterministic with more than a third of flaky tests (38.1%) while `SpatialVLA` only has two of such tests. The results are similar for the different number of distractor objects: it seems that there is no correlation between the level of non-determinism and the number of distractor objects.

Fig. 3 reports the FPD for each model grouped by the RSR according to Table I. For `GR00T N1` (Fig. 3a), we observe that when the test is successful across repetitions (i.e., *All passing*), the final position of the target object is rather consistent, with slight variations in a minority of cases. For tests in which all repetitions fail (i.e., *None passing*), most of them are consistent but there are still some tests where small and large variations happen. The largest variations occur when the result is not consistent (i.e., *2 passing* and *1 passing*).



(a) `GR00T N1` (b) `SpatialVLA`

Fig. 4: RQ1.c – Distribution of FPD across AL ranges



(a) `GR00T N1` (b) `SpatialVLA`

Fig. 5: RQ2 – Distribution of SPD by simulation step

For `SpatialVLA` (Fig. 3b), we observe that FPD is systematically close to 0 for deterministic cases (i.e., *All passing* and *None passing*). From Table I, we observe that two tests have results differing across repetitions; for these, in Fig. 3b, we observe a high FPD, meaning there is still some level of non-determinism in the model.

3) RQ1.c: Fig. 4 reports, for the two models, the FPD relative to the AL (see Sect. II-E3). For `GR00T N1` (Fig. 4a), we see that on average, the FPD is getting higher and more dispersed when the tests take longer to succeed. However, we find that when the test did not succeed (i.e., $AL = 121$), the median FPD is quite low (yet there is high dispersion in both directions). For `SpatialVLA` (Fig. 4b), as successes and failures are fairly deterministic, we observe that the FPD remains at 0 across all lengths.

> **Answer to RQ1**: The two VLAs are non-deterministic to very different degrees. `GR00T N1` can either succeed or fail at the same test. For uncertain tests, the final state of the simulation may vary by either a small or high degree. Tests having longer executions exhibit more non-determinism. For `SpatialVLA`, we find the outcomes to be extremely consistent but still not fully deterministic.

B. Answer to RQ2

In this RQ, we will evaluate non-determinism throughout the execution. Fig. 5 shows the distribution of SPD relative to simulation step (see Sect. II-E4) for the entire test suite.

In the case of `GR00T N1` (see Fig. 5a), we find that SPD is near 0 on the first 10 steps of the simulation, before starting to increase from step 15 onwards until step 30. From step 30 onwards, we observe that the median SPD and dispersion remain more or less stable, hinting that executions cannot deviate more than a certain degree, which in turn limits the non-determinism.

As for `SpatialVLA` (see Fig. 5b), SPD is very close to 0 ($\approx 10^{-16}$) for every step, with minor variations not visible in the figure. Once again, this suggests almost deterministic behaviour for this model.

> **Answer to RQ2**: In the case of `GR00T N1`, simulations are nearly deterministic at the start before the deviations start to grow at around 10 steps. SPD stabilizes at some point, suggesting that repetitions can only differ by a certain degree. For `SpatialVLA`, we see almost no deviations.

IV. Discussion and Conclusion

In this work, we evaluated two VLAs in simulated environments to assess the degree of non-determinism in their executions. Our results show that:

- Models are deterministic to very different degrees, with `SpatialVLA` being extremely consistent and `GR00T N1` showing a high degree of non-determinism.
- Divergence in non-deterministic executions gradually grows at the beginning of the test but starts stabilizing, indicating that deviations exist within a limited range for those executions.

Our recommendation to the VLA testing community, based on our study, is to carefully evaluate the non-determinism of subject VLA models and ensure sufficient repetitions of tests are done to ensure reliability of results. Future work may increase the number of subject VLAs to establish a classification of more and less deterministic models, and evaluate non-determinism on more criteria than the target object location, such as coordinates of the robotic embodiment's gripper or joints.

References

[1] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. GR00T N1: an open foundation model for generalist humanoid robots. *CoRR*, abs/2503.14734, 2025.

[2] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[3] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, S. Levine, J. Wu, C. Finn, H. Su, Q. Vuong, and T. Xiao. Evaluating Real-World Robot Manipulation Policies in Simulation, May 2024. arXiv:2405.05941 [cs].

[4] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, and X. Li. SpatialVLA: Exploring spatial representations for visual-language-action model. *CoRR*, abs/2501.15830, 2025.

[5] A. Rasouli, M. Alban, S. Pakdamansavoji, Z. Li, Z. Zhang, A. Wu, and X. Zhao. Distracted robot: How visual clutter undermine robotic manipulation. *arXiv preprint arXiv:2511.22780*, 2025.

[6] R. Shao, W. Li, L. Zhang, R. Zhang, Z. Liu, R. Chen, and L. Nie. Large VLM-based vision-language-action models for robotic manipulation: A survey. *CoRR*, abs/2508.13073, 2025.

[7] P. Valle, C. Lu, S. Ali, and A. Arrieta. Evaluating uncertainty and quality of visual language action-enabled robots. *CoRR*, abs/2507.17049, 2025.

[8] P. Valle, S. Segura, S. Ali, and A. Arrieta. Metamorphic testing of vision–language action–enabled robots. In *2026 IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 52–63, 2026.

[9] Z. Wang, Z. Zhou, J. Song, Y. Huang, Z. Shu, and L. Ma. VLATest: Testing and evaluating vision-language-action models for robotic manipulation. *Proc. ACM Softw. Eng.*, 2(FSE), June 2025.