Nouns, Not Modifiers: OpenVLA Parses Objects but Fails at Spatial Composition

Jin Yoo [1]

Abstract

Vision-Language-Action models are trained predominantly on absolute instructions that name specific objects, but real-world deployment demands the capacity to understand compositional instructions that combine object identifiers with spatial modifiers. We evaluate OpenVLA-7B on relational instructions using a matched-pair design in SimplerEnv by testing six instruction types on 20 scenes: two absolute, two distance-based, and two spatial. OpenVLA demonstrates 90.0% accuracy on absolute instructions but fails at chance on distance (52.5%) and spatial (47.5%) instructions, with 15% accuracy on 'left' and 80% accuracy on 'right' revealing a systematic rightward default. Failure mode analysis reveals that 93% of failures involve the end-effector reaching for the wrong can, rather than the third distractor object, which shows that the model correctly parses the object category but fails to compose it with the relational modifier. This paper shows that compositional spatial reasoning does not emerge in OpenVLA-7B on SimplerEnv MoveNear, and identifies key failure patterns that must be addressed prior to VLA deployment in the real world.

1. Introduction

Compositional generalization, which is the ability of AI models to understand novel combinations of known, simpler primitives, is a critical frontier in machine learning (Lake & Baroni, 2018). While models trained on the command "pick up the coke can" and "the can on the left" may interpret the individual phrases, combining the two concepts into understanding "pick up the can on the left" requires compositional reasoning that may not emerge from training data alone.

[1]Duke University, Durham, NC, USA. Correspondence to: Jin Yoo <jin.yoo@duke.edu>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

VLA models have shown rapid progress in robotic manipulation. Models like OpenVLA (Kim et al., 2025) are trained on large-scale datasets that have a focus on absolute instructions ("pick up the coke can"), and language grounding is viewed as an emergent property from massive training data. In real-world scenarios, however, VLAs may encounter prompts such as "pick up the can on the left" or "move the closer bottle," which requires identifying object nouns while handling spatial-relational modifiers and grounding them in the visual scene. Whether current VLAs can handle such compositional tasks remains untested.

In this paper, we test OpenVLA-7B on relational instructions using a matched-pair experimental design. We reveal that OpenVLA performs at chance on relational instructions despite near-ceiling accuracy on absolute instructions, and identify systematic directional biases and failure modes that characterize this compositional gap.

2. Related Work

Recent advancements in language and vision-conditioned robotics have enabled integration between language comprehension capacity and physical robotic control. Ahn et al. (2022) pioneered the usage of Large Language Models for high-level action planning in response to a task, combining the LLM's semantic knowledge with learned affordance functions. Low-level task execution, however, was kept separate for independently trained policies. Zitkovich et al. (2023) unified this pipeline, co-fine-tuning a Vision Language Model to directly output action tokens in text, which demonstrated emergent semantic reasoning and generalization capacity to novel objects. Kim et al. (2025) made these advancements accessible as an open-source, 7-billion-parameter model. Across this progression, language grounding has been treated as an implicit emergent capacity derived from large-scale, diverse training data.

Pugacheva et al. (2025) analyzed the effects of linguistic perturbations in instructions to multiple VLA models, including OpenVLA. The researchers tested model robustness against variations in instructions from human-generated paraphrasing and appended irrelevant context, and discovered performance drops of approximately 50% when appended context

is lexically and semantically similar to the original instructions and of matching length. Our work seeks to address a complementary question: instead of testing the model's capacity to filter out noise, we test if OpenVLA is capable of handling relational instructions that require composing object nouns with spatial modifiers. Furthermore, our failure mode decomposition and directional bias analysis characterize how the model fails.

Shin (2026) addresses OpenVLA's limited linguistic generalization capabilities by generating semantically equivalent instructions with syntactic diversity using a Large Language Model. Afterward, the researchers use LoRA to fine-tune the model to increase robustness to phrasing variations and show that increased linguistic diversity in training data improves generalization capabilities. We seek to explore deeper forms of linguistic generalization by creating compositionally novel instructions where the correct target changes depending on a relational modifier that is grounded in the visual scene.

Qu et al. (2025) argues that spatial understanding of a visual scene is a key bottleneck in VLA performance. In response, the researchers propose the injection of 3D spatial representations into VLA architecture. After being trained on 1.1 million real-world episodes, SpatialVLA indicates improved zero-shot spatial reasoning across various robot platforms. Similarly, Zhang et al. (2025) addresses this gap by injecting 3D spatial tokens from spatial foundation models into the action head. While their work focuses on proposing an architectural improvement in spatial reasoning failures, we provide a controlled diagnostic of the failure itself. By holding the visual scenes constant while varying only the instruction structure, we identify specific patterns in how OpenVLA fails on spatial-relational instructions.

3. Method

We evaluate OpenVLA-7B (Kim et al., 2025), a seven billion parameter, open-source Vision-Language-Action model trained on 970k real-world robot demonstrations from the Open X-Embodiment dataset. We use the original, pretrained model without fine-tuning, and run greedy inference using the built-in `predict_action` method with the `fractal20220817_data` unnormalization key. Each trial includes 40 inference steps, after which we record the final position of the end-effector (EE). We selected 40 steps based on a calibration that tracked EE movement at 10-step intervals. This revealed that the arm position converges to a stable position by 20-30 steps on absolute instructions, with negligible displacement thereafter. Left and right are defined from the overhead camera's perspective, which matches the robot's operational frame. The can with lower Y coordinate in the scene is on the visual left.

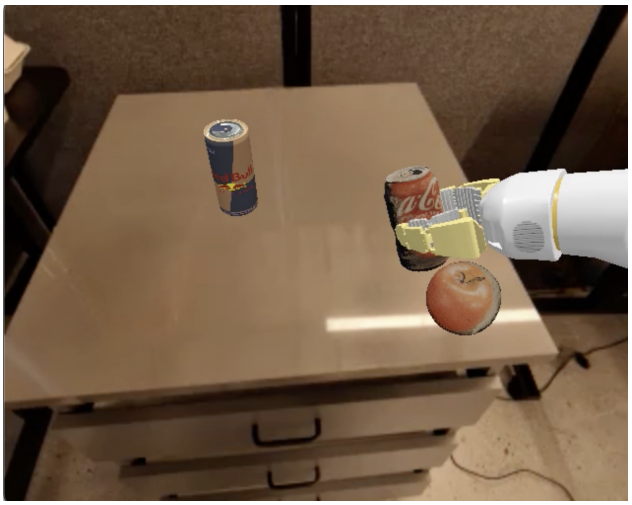

Figure 1. Example scene from our experiment. Two cans (redbull, left; coke, right) and one distractor (apple) are placed on a table, and the end effector grabs the coke. The same scene is presented with six different text instructions, and the final end-effector position determines which object was selected.

All trials were conducted within the SimplerEnv simulation framework (Li et al., 2024), which utilizes ManiSkill2 and the SAPIEN physics engine. We use the `MoveNearGoogleBakedTexInScene` environment with a Google Robot Static Manipulator, simulated at 513 Hz with a control frequency of 3 Hz. At each step, we use segmentation masks to overlay real-world RGB images onto the simulation, which ensures the visual input will resemble the real-world images OpenVLA was trained on. The environment provides two distinct object triplets, with two cans each. The first (episode 24) contains coke, redbull, and an apple. The second (episode 48) contains pepsi, redbull, and an orange.

We construct 20 custom scenes from the two triplets by repositioning objects using SAPIEN's `set_pose` API. We generate five distinct position layouts that define where the two cans will be placed on the table. Each of these layouts is applied four times, once for each of the two scene types (coke + redbull + apple, pepsi + redbull + orange), and once per left-right swap (which can is placed on the left/right position). This produces a total of 20 scenes ($5 \times 2 \times 2$). Across these 20 scenes, redbull appears on the left in 10 scenes and on the right in 10 scenes. The left can is closer to the camera in 10 scenes and farther in 10. A third non-can object (apple, orange) is placed away from both cans in all scenes.

For every scene, we test six natural language instructions: two absolute ("pick up the coke can," "pick up the redbull can"), two distance-based ("pick up the can closer to you," "pick up the can farther from you"), and two spatial ("pick up the can on the left," "pick up the can on the right").

Table 1. Primary accuracy by instruction category. All p-values are from binomial tests against the 50% chance baseline, Bonferroni-corrected ($n = 3$).

Type	Accuracy	95% CI	Bonf. p
Absolute	90.0% (36/40)	[76.9, 96.0]	<0.001***
Distance	52.5% (21/40)	[37.5, 67.1]	1.000
Spatial	47.5% (19/40)	[32.9, 62.5]	1.000

Table 2. Subtype breakdown. Left vs. Right: Fisher's exact Bonferroni-corrected $p < 0.001$. Closer vs. Farther: $p = 1.0$.

Subtype	Accuracy	95% CI
Absolute (A)	80.0% (16/20)	[58.4, 91.9]
Absolute (B)	100.0% (20/20)	[83.9, 100.0]
Closer	60.0% (12/20)	[38.7, 78.1]
Farther	45.0% (9/20)	[25.8, 65.8]
Left	15.0% (3/20)	[5.2, 36.0]
Right	80.0% (16/20)	[58.4, 91.9]

This generates 120 trials (20×6). A critical detail is that the environment resets the object positions and the robot's physical state in every trial, leaving only the text prompt variable. This means that every group of 6 instructions is tested on an identical visual scene, creating a matched-pairs design that ensures that any difference in behavior across instruction types is attributable to the instruction text rather than visual confounds.

After each trial, we compute a Euclidean XY distance from the final end-effector position to each of the three objects. Our primary outcome metric measures whether the EE was closer to the correct can than the other can, which creates a binary correct/incorrect measurement with a random chance baseline of 50%. Our secondary metric measures whether the closest object, including all three objects, to the EE was the correct can, which produces a 33% chance baseline. We also classify each incorrect trial by which of the three objects was nearest to the final EE position.

4. Results

We report 95% Wilson score confidence intervals for all proportions. All p-values are Bonferroni-corrected within each family of comparisons.

As shown in Table 1, OpenVLA demonstrated 90.0% accuracy on absolute instructions, significantly above the 50% chance baseline (Bonferroni-corrected $p < 0.001$). The model's performance on distance and spatial instructions was indistinguishable from chance (52.5% and 47.5%, both $p = 1.0$). Pairwise Fisher's exact tests verified that absolute accuracy was significantly higher than distance (OR $= 8.14$, Bonferroni-corrected $p = 0.001$) and spatial (OR $= 9.95$, $p < 0.001$). Distance and spatial accuracy did not significantly differ (OR $= 1.22$, $p = 1.0$), indicating a common compositional processing failure across both relational instruction types. The two absolute subtypes did not differ significantly (80% vs. 100%, Bonferroni-corrected $p = 0.33$), showing that the model reliably identified the correct can regardless of the target. Because trials share scenes across instruction types, we re-verified our comparisons using McNemar's test for paired data, which yielded consistent significance patterns (e.g., left vs. right: $p = 0.002$).

Table 2 reveals a critical directional asymmetry within spatial instructions. "Pick up the can on the right" succeeded 80% of the time, while "pick up the can on the left" had 15% accuracy (Fisher's exact Bonferroni-corrected $p < 0.001$). Unlike spatial instructions, the two distance subtypes showed no significant difference (60% vs. 45%, Bonferroni-corrected $p = 1.0$). This discrepancy indicates contrasting failure mechanisms for the two relational categories. Distance instructions fail symmetrically, indicating a general inability to process egocentric depth modifiers, though the non-significant closer vs. farther result should be interpreted with caution given limited power at $n = 20$ per subtype. Spatial instructions fail asymmetrically, with the model defaulting to the rightward object regardless of the instruction, suggesting a systematic rightward prior. The current design cannot distinguish whether the model processes spatial modifiers and misapplies them due to distortions in spatial representations, or ignores the modifier entirely in favor of positional defaults.

To rule out additional alternative explanations, we analyzed can-pair bias and failure modes. No significant difference in accuracy was detected between coke + redbull and pepsi + redbull scenes for any instruction category (all $p = 1.0$), indicating that results were not driven by a preference for specific cans. Of the 44 incorrect trials, 93% involved the EE reaching toward the wrong can rather than the non-can distractor, with a mean distance of 0.09m to the wrong can versus 0.21m to the distractor, confirming that the EE committed to the wrong can rather than failing to reach a more distant distractor. Within spatial instructions specifically, all 21 failures were caused by reaching for the wrong can. This shows that the model correctly parses "can" but fails to apply the relational modifier to distinguish between the two can objects. The four absolute failures on redbull were distributed between the distractor (2 trials) and the other can (2 trials), suggesting no systematic redbull-specific confusion.

5. Discussion and Conclusion

Our results indicate that OpenVLA has shallow language grounding. While the model correctly maps object names to motor actions, it fails to compose those names with given spatial modifiers to resolve referential ambiguity. This is not a product of visual confounds—the model has access

to the same scene from the same camera angle for every instruction type. Instead, it is a result of a compositional generalization gap in how the model handles instruction text. The Open X-Embodiment training data (O'Neill et al., 2023) is dominated by absolute instructions that name specific objects (e.g., "pick up the coke can," "place the sponge on the plate"), offering little explicit supervision for relational language grounding. This demonstrates that OpenVLA-7B fails to acquire compositional spatial reasoning in this evaluation setting.

The 15% vs. 80% left/right asymmetry, with 100% of spatial instruction failures caused by the EE reaching toward the wrong can, is especially informative. The rightward default was consistent across all five base layouts, with "left" accuracy at or below 25% in every configuration. Moreover, across all spatial trials, the EE moved to the physically right can regardless of instruction word in $33/40$ trials (82.5%). This indicates that OpenVLA fails on spatial instructions by defaulting to the rightward can almost unconditionally. This rightward default may share a common origin with the left-first spatial attention bias documented in Vision-Language Models (Chaudhary et al., 2025), where models describe left-positioned content before right-positioned content. Because a left-to-right processing order attends to the rightmost object last, recency in autoregressive action decoding could surface that object as the default target, resulting in an action-space counterpart to their left-first *description* bias rather than a direct copy of it. The lack of a comparable asymmetry in distance-based prompts (60% vs. 45%, $p = 1.0$) reveals different failure mechanisms: spatial instructions trigger a learned positional default, while distance instructions produce undirected, chance-level behavior.

Several factors constrain the scope of these findings. Our experiment is limited to OpenVLA-7B in simulation with 120 trials, meaning generalization to other VLAs such as RT-2 or π_0, or to real-world scenarios, is untested. Redbull appears in every scene due to SimplerEnv's limited object selection; while counterbalancing mitigates this, a fully balanced object pool would strengthen the design. Future work should test whether fine-tuning on relational demonstrations closes this gap, whether directional bias persists across multiple VLA architectures, and whether 3D spatial representations such as those proposed by Qu et al. (2025) can address the compositional bottleneck we identify.

Impact Statement

We identify a safety-relevant limitation in VLA language grounding. Current models may fail silently on relational instructions, resulting in the robot reaching for the wrong object with high confidence. Deploying VLAs in contexts where users issue spatial-relational commands without awareness of this limitation could lead to incorrect object manipulation. We recommend that practitioners verify VLA instruction-following capabilities on relational prompts before deployment.

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., and Zeng, A. Do as I can, not as I say: Grounding language in robotic affordances. In *Conference on Robot Learning*, 2022.

Chaudhary, A., Goyal, S., Narang, P., and Kumar, D. Investigating spatial attention bias in vision-language models. *arXiv preprint arXiv:2512.18231*, 2025.

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E. P., Sanketi, P. R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., and Finn, C. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270, pp. 2679–2713. PMLR, 2025.

Lake, B. M. and Baroni, M. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *Proceedings of the 35th International Conference on Machine Learning*, 80:2873–2882, 2018.

Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H. R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., Levine, S., Wu, J., Finn, C., Su, H., Vuong, Q., and Xiao, T. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

O'Neill, A. et al. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023.

Pugacheva, D., Moskalenko, A., Shepelev, D., Kuznetsov, A., Shakhuro, V., and Tutubalina, E. Bring the apple, not the sofa: Impact of irrelevant context in embodied AI commands on VLA models. *arXiv preprint arXiv:2510.07067*, 2025.

Qu, D., Song, H., Chen, Q., Yao, Y., Ye, X., Ding, Y., Wang, Z., Gu, J., Zhao, B., Wang, D., et al. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems*, 2025.

Shin, D. Enhancing linguistic generalization of VLA: Fine-tuning OpenVLA via synthetic instruction augmentation. *arXiv preprint arXiv:2603.16044*, 2026.

Zhang, Z., Li, H., Dai, Y., Zhu, Z., Zhou, L., Liu, C., Wang, D., Tay, F. E. H., Chen, S., Liu, Z., Liu, Y., Li, X., and Zhou, P. FALCON: From spatial to actions: Grounding vision-language-action model in spatial foundation priors. *arXiv preprint arXiv:2510.17439*, 2025.

Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P. R., Salazar, G., et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229, pp. 2165–2183. PMLR, 2023.