Bring the Apple 🍎, Not the Sofa 🛋️: Impact of Irrelevant Context in Embodied AI Commands on VLA Models

Andrey Moskalenko[1,2,3*], **Daria Pugacheva**[1,5*], **Denis Shepelev**[1,3]
Andrey Kuznetsov[1,6], **Vlad Shakhuro**[1,2,3], **Elena Tutubalina**[1,5,7]
[1]AIRI, [2]Lomonosov Moscow State University, [3]NUST MISIS,
[4]IAI MSU, [5]HSE University, [6]Innopolis University, [7]Sber AI
Correspondence: dpugacheva@hse.ru, amoskalenko@fusionbrainlab.com, tutubalina@airi.net

Abstract

Vision Language Action (VLA) models are widely used in Embodied AI, enabling robots to interpret and execute language instructions. However, their robustness to natural language variability in real-world scenarios has not been thoroughly investigated. In this work, we present a novel systematic study of the robustness of state-of-the-art VLA models under linguistic perturbations. Specifically, we evaluate model performance under two types of instruction noise: (1) human-generated paraphrasing and (2) the addition of irrelevant context. We further categorize irrelevant contexts into two groups according to their length and their semantic and lexical proximity to robot commands. In this study, we observe consistent performance degradation as context size expands. We also demonstrate that the model can exhibit relative robustness to random context, with a performance drop within 10%, while semantically and lexically similar context of the same length can trigger a quality decline of around 50%. Human paraphrases of instructions lead to a drop of nearly 20%. Our results highlights a critical gap in the safety and efficiency of modern VLA models for real-world deployment.

1 Introduction

Embodied AI is undergoing rapid development, with robotic systems increasingly exhibiting practical utility in everyday environments. Vision-Language-Action (VLA) models play a central role in enabling this progress. By leveraging large language models (LLMs), robots can interpret and execute natural language instructions grounded in visual perception (Collaboration et al., 2023; Jiang et al., 2023; Driess et al., 2023; Zhou et al., 2025).

Despite this momentum, deploying VLAs outside curated lab conditions exposes a persistent fragility: real users rarely issue instructions in the single "canonical" phrasing used during training or benchmarking. Instead, commands are naturally paraphrased, embedded in longer utterances, and often accompanied by irrelevant details, e.g., explanations, side remarks, or surrounding conversation (Figure 1). Such linguistic variability is not a corner case, it is the default interaction mode in homes, offices, and retail settings. Yet most VLA evaluations still assume short, clean, task-focused prompts (Liu et al., 2023; Lynch et al., 2023; Nasiriany et al., 2024), leaving a gap between reported benchmark performance and the linguistic conditions encountered in practice.

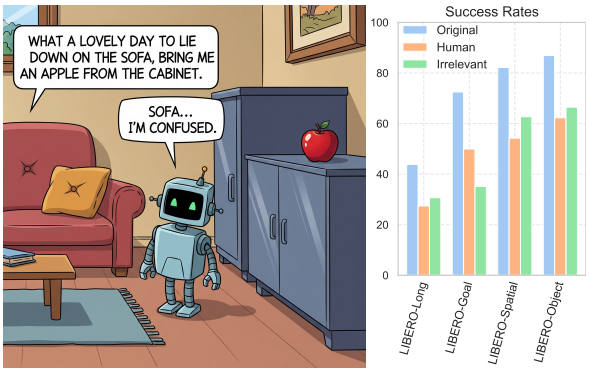


Figure 1: Human-voiced commands to the robot may contain irrelevant context and cause the target command to fail. We observed a significant drop in the success rates of VLA robotic models when real users posed problems.

This gap raises a simple but consequential question: *how robust are state-of-the-art VLA models to realistic instruction noise?* Most embodied instruction-following work still reports overall task success under templated/crowdsourced directives (Shridhar et al., 2020; Collaboration et al., 2023; Octo Model Team et al., 2024), leaving language variability largely under-explored in standard evaluation protocols. Meanwhile, robustness efforts in robotics often emphasize visual and action perturbations (Wang et al., 2024a;

*Equal contribution.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 4: Student Research Workshop, pages 840–860

Zhang et al., 2025) rather than systematic natural-language variation. Although a few recent studies explicitly probe instruction diversity and paraphrastic robustness in embodied settings, they typically do so in specific task families or under limited perturbation designs, and do not yet provide a comprehensive picture for modern VLA instruction-following under real conversational noise (Parekh et al., 2024; Szot et al., 2024). However, for embodied agents, small changes in instruction form can alter goal interpretation, grounding, or action sequencing, so benign linguistic variation can induce failures that are practically as costly as perception errors.

To address these gaps, we introduce an evaluation protocol for measuring VLA robustness to realistic instruction perturbations. We focus on two prevalent noise sources: *irrelevant context* and *paraphrasing*. For irrelevant context, we construct a controlled suite of distractors that varies along two axes: (i) **length**, to quantify how performance changes as non-actionable context grows, and (ii) **semantic/lexical proximity** to the target command, to distinguish benign random chatter from confusable, instruction-like distractors. In addition, we collect human-written paraphrases for each instruction in our benchmark to study robustness under natural rephrasings that preserve intent.

We perform evaluations using two well-known simulation benchmarks, LIBERO (Liu et al., 2023) and Habitat 2.0 (Szot et al., 2021). Our study covers five state-of-the-art VLA models: OpenVLA (Kim et al., 2025), UniAct (Zheng et al., 2025), MoDE (Reuss et al., 2025), π_0 (Black et al., 2024), and LLARP (Szot et al., 2024).

Overall, our contributions are as follows:

- We evaluate VLA models for various embodiments and identify that these models are most vulnerable to irrelevant context, which is lexically and semantically close to the commands from the training set.
- We show that the performance degrades as the length of irrelevant context increases and can drop by up to 58%, when the context length approaches the length of target commands.
- We perform a human study and show that natural paraphrasing drops VLA model performance by 20%, revealing adaptation gaps between LLM-based VLA models and real-world deployment needs.

2 Related Work

VLA models enable robots to take visual observations and natural language commands as input and output low-level actions for control. We focused on the task of assessing robustness of these models to linguistic variation the ability to understand paraphrased or syntactically altered commands that were not seen during training, which is crucial for real-world applications of these models.

2.1 VLA Models

Recent advances in VLA models have demonstrated the integration of web-scale multimodal pretraining with robotic control through co-fine-tuning of vision-language models on robot trajectory datasets.

RT-1 (Brohan et al., 2023) was a pioneering VLA-like model for real-world robotic manipulation. It processes a short sequence of camera images together with a task description in natural language, and outputs a sequence of robot actions. RT-2 (Zitkovich et al., 2023) exhibits emergent semantic reasoning and generalization to novel objects and instructions by encoding actions as text tokens alongside natural language.

Significant progress in the field has occurred with the release of the open-source OpenVLA (Kim et al., 2025) foundation model, which explicitly integrates a large language model to strengthen language understanding. OpenVLA is a 7B policy built on a Llama 2 (Touvron et al., 2023) model, fused with vision encoders for image input. It was trained on 970k real robot demonstrations (Collaboration et al., 2023) drawn from diverse sources, as well as additional Internet-scale vision-language data to inject world knowledge. Due to its openness, this model became the basis for subsequent work in this area (Black et al., 2024; Belkhale and Sadigh, 2024; Wen et al., 2025; Qu et al., 2025; Zheng et al., 2025; Reuss et al., 2025; Lykov et al., 2025). Moreover, this approach was also utilized in drone control (Lykov et al., 2025; Serpiva et al., 2025) and autonomous vehicles (Arai et al., 2025; Zhou et al., 2025). Thus, due to the significant growth of popularity of the models of the VLA family, we are conducting our research to understand the robustness of

such models to the variability of text prompts.

2.2 Evaluation in Simulation Environments

As a rule, robotics models are usually evaluated using success rate (SR) in a simulator and real world environments. We believe it would be unsafe to evaluate deviant robotic behavior in the real world, so we focus mainly on simulator environments. Unlike the real world, simulators allow accurate reproduction of all initial states, so different models can be compared objectively. Thus, simulation environments have become indispensable for systematically benchmarking robotics models under controlled yet diverse conditions.

There are many simulation environments available. RoboCasa (Nasiriany et al., 2024) is a simulation framework for training generalist robots in realistic home environments. SimplerENV (Li et al., 2024b) offers a suite of simulated replicas of common real-robot setups, enabling scalable, reproducible evaluation and demonstrating strong correlation with real-world performance for generalist policies.

Habitat (Savva et al., 2019) is a high-performance simulator for embodied AI and navigation tasks, capable of rendering RGB-D observations and simulating rigid-body dynamics at over 8,000 steps per second in photorealistic 3D scenes.

LIBERO (Liu et al., 2023) provides a lifelong learning benchmark with procedurally generated manipulation tasks, specifically designed to study declarative and procedural knowledge transfer in simulation at scale. LIBERO is organized into four distinct task suites designed to probe different facets of lifelong learning in robot manipulation.

We mainly focused on Habitat and LIBERO for our experiments, since they are now popular simulation environments to benchmark VLA models.

2.3 VLA Robustness

Robustness is an active area of evaluation for VLA models. Wang et al. (2024a) presented a study of adversarial attacks on Vision-Language-Action models, highlighting novel vulnerabilities unique to robotic control tasks. They introduce two attack objectives: an untargeted position-aware attack that perturbs spatial inputs to destabilize controller outputs and a targeted manipulation attack that crafts minimal perturbations to redirect robot trajectories toward specific failure modes. However, the authors study only image perturbation robustness at the robot's input, which is a rarer case because the robot's camera is inside it and can only be attacked with physically printed patches. We study resistance specifically to text prompts because the user always has direct influence on them.

Recent comparative studies have explicitly tested a number of models on paraphrased or altered instructions to probe their robustness. LADEV (Wang et al., 2024b) is a language-driven evaluation framework that generates paraphrases of task instructions (using LLMs generation method) to test VLA policies. Researchers compared multiple models on the same set of tasks under original and paraphrased commands. Szot et al. (2024) investigate the robustness of their proposed model to paraphrasing and irrelevant context, but their analysis is restricted to a limited set of templates, i.e. one for irrelevant context and four for paraphrasing. Similarly, Parekh et al. (2024) focus solely on template-based paraphrasing, but does not examine the influence of irrelevant context. Moreover, this work does not consider how real users might naturally paraphrase task instructions. We extend this work by using a simulator with a larger number of robotic tasks, as well as we also proposed intelligent generation of text paraphrases of different categories, and also showed how to improve the robustness of models to such reformulations.

3 Evaluation Setup for VLA Models

In this section, we detail the experimental setup used to systematically benchmark the robustness of VLA models to linguistic perturbations. We begin with introducing the simulation environments 3.1 and VLA models 3.2 used in our study. Next, we propose several types of irrelevant context 3.3 and present crowdsourced paraphrases of robot commands 3.4 to assess model robustness. Finally, we report experimental results and provide their analysis 4.

3.1 Simulation Environments

We study the robustness of the VLA models in the LIBERO (Liu et al., 2023) and Habitat 2.0 (Szot et al., 2021) simulation environments.

LIBERO (Liu et al., 2023) is designed to evaluate models on object manipulation tasks. Each LIBERO task suite focuses on a specific type of distribution shift or knowledge transfer challenge, enabling controlled evaluation of model capabilities under spatial, object, goal, and entangled task

Environment	Variation		Command
Habitat 2.0 		Original	Find an orange and move it to the sink.
		Human	Can you find an orange and put it in the sink?
	Context Length	Single	*Although,* find an orange and move it to the sink.
		Short	*Inspired while cooking dinner.* Find an orange and move it to the tv stand.
		Long	*He felt motivated cleaning the pantry and organizing everything, so* find an orange and move it to the sink.
	Context Semantic	Location	*There's an apple on the TV stand, but* find an orange and move it to the sink.
		Description	*Cup is a container for liquids.* Find an orange and move it to the sink.
		Infeasible	*Bake a pie with peach slices.* Find an orange and move it to the sink.
LIBERO 		Original	put the wine bottle on top of the cabinet
		Human	move the bottle of wine to the top of the cabinet
	Context Length	Single	*moreover* put the wine bottle on top of the cabinet
		Short	*nostalgia strikes after dinner* put the wine bottle on top of the cabinet
		Long	*the gloomy weather matched her tired and melancholy* put the wine bottle on top of the cabinet
	Context Semantic	Location	*the bowl is in the basket* put the wine bottle on top of the cabinet
		Description	*padlock are made of metal* put the wine bottle on top of the cabinet
		Infeasible	*bite into the soft plum* put the wine bottle on top of the cabinet

Table 1: Examples of context inserted into commands for the Habitat 2.0 simulator and LIBERO benchmark.

variations. We consider the following LIBERO task suites:

- LIBERO-Spatial: contains 10 short-horizon tasks that require the robot to transfer and memorize new spatial relationships.
- LIBERO-Object: comprises 10 short-horizon tasks centered on learning new object types, where the robot must pick and place different objects in sequence.
- LIBERO-Goal: includes 10 short-horizon tasks that share identical objects and spatial layouts but differ only in procedural goals, testing the transfer of motion and behavior knowledge.
- LIBERO-Long (also called LIBERO-10) comprises 10 long-horizon tasks, reserved for downstream evaluation of lifelong learning algorithms.

Habitat 2.0 (Szot et al., 2021) is a simulation platform that supports not only object manipulation but also navigation tasks. Following the authors' instructions (Szot et al., 2024), we generated 100 language commands for evaluation. Both the generated commands and those from the training set included punctuation marks and letters in various cases, such as, "Find an apple and put it away in the fridge." Moreover, these commands could also be phrased as questions, offering a greater diversity compared to the commands found in LIBERO.

3.2 VLA Models

In LIBERO, we evaluate three state-of-the-art and popular models: OpenVLA (Kim et al., 2025), UniAct (Zheng et al., 2025), Mixture-of-Denoising Experts (MoDE) (Reuss et al., 2025), π_0 (Black et al., 2024). In Habitat 2.0, we evaluate LLARP model (Szot et al., 2024). The OpenVLA and LLARP models are built upon Llama 2 7B (Touvron et al., 2023) LLM backbone, PaliGemma 3B (Beyer* et al., 2024) with Gemma 2B LLM backbone was used for π_0. UniAct is a lightweight model based on LLaVA-OneVion-

0.5B (Li et al., 2024a) with Qwen 2 backbone and performance exceeded OpenVLA. MoDE leverages a frozen CLIP language encoder. To ensure lower variance in the experimental results, models are evaluated on LIBERO benchmarks across 50 trials for each task suite, and the reported performance is the average success rate over three random seeds (resulting in 150 total trials per statistic).

During the rollout phase of LLARP, the policy acts in parallel in 32 Habitat 2.0 environments and are evaluated across 30 trials for each task, and the reported performance is the average success rate over three random seeds as well.

3.3 Irrelevant Context

We consider several types of irrelevant context and organize them into two groups: (1) context length variation, (2) semantic and lexical similarity.

The first group of contexts was chosen to be lexically and semantically different from the commands of the training set, and varied in length. The context from the second group contained names of scene objects and constructions similar to the training commands. All contexts are generated using GPT 4.1 and then verified by experts. Each context is added both before the target command and afterward. We adapt the final noisy command to maximize similarity to the template from the model training set in order to eliminate the possible impact of punctuation and letter case changes (please see Table 1 with examples).

Context length variation Specifically, the first set consists of a context *"Single"*, which includes single introductory word like 'However', 'Moreover' etc; contexts *"Short"* and *'Long'* includes 3-5 or 7-10 words sentences whose content represented random phrases unrelated to the roboarm commands or objects in the scene, e.g., 'the weather is nice today' or 'the gloomy weather matched her tired and melancholy mood today'.

Semantic and lexical similarity The second set also comprises three types of context.

The first type of context *"Description"* provides semantic proximity to the training set. It contains short phrases describing the random object of the scene, but this description was arbitrary. It did not include information about the location of the object or the action to be performed with the object, e.g. "Cup is a container for liquids. Find an orange and move it to the TV stand".

The next type *"Infeasible"* represents infeasible commands, which the roboarm cannot execute, and which did not occur in the training set, e.g., "Bake a pie with peach slices. Find an orange and move it to the right counter". It is semantically and grammatically close to training commands, but differs lexically.

Finally, the last type *"Location"* combines both semantic and lexical proximity to what the model observed in training. It consists of short phrases with 3-5 words that contain references to the location of the objects in the scene. The location and the names of the objects themselves did correspond to the content of the scene, but the subsequent command was not related to the object, e.g., "There's an apple in the cabinet, but find a screwdriver and move it to the left counter." A more complete list of examples for each type of context can be found in the Appendix A.1.

For each target command, context was injected both before and after the command. We provide averaged results for these two injection types.

3.4 Command Paraphrasing

To evaluate the robustness of VLA models to command paraphrasing, we conducted a real-user study. Specifically, crowdworkers were asked to paraphrase task descriptions drawn from experimental simulation benchmarks. All commands were originally written in English, so we restricted participation to workers who passed an English-proficiency test. To avoid introducing annotation bias, instructions were kept as minimal as possible, with the sole requirement that the reformulated text preserve the meaning of the original. Participants saw the instruction from Figure 2.

Each worker received a batch of five descriptions per task and spent on median 296 seconds (including instruction time) to complete the task. Each description was independently paraphrased by five different crowdworkers.

All collected paraphrases were then reviewed by our in-lab experts, who retained only those submissions in which the semantic content of the original description was faithfully preserved.

The resulting texts were then used to evaluate the performance of the VLA models by replacing the original task prompts in the simulation benchmarks with texts formulated by real-users.

Environment	Model	Original	Length			Semantic			Paraphrasing	
			Single	Short	Long	Description	Infeasible	Location	Human	DeepSeek
LIBERO Goal	OpenVLA	77.5	67.6	43.5	**18.9**	30.4	28.0	**25.5**	58.2	**54.8**
	UniAct	67.5	62.5	39.5	**28.5**	30.5	28.3	**16.0**	41.7	**38.8**
	π_0	91.5	91.6	77.9	**44.8**	68.8	59.6	**55.6**	78.5	**71.2**
LIBERO Object	OpenVLA	87.3	86.3	74.2	**56.2**	70.3	**62.5**	72.5	**80.0**	82.8
	UniAct	86.5	82.0	64.0	**47.0**	59.8	**55.3**	63.8	**44.6**	58.2
	π_0	97.5	97.4	94.8	**84.9**	91.8	92.5	**85.9**	**89.7**	95.8
LIBERO Spatial	OpenVLA	85.3	82.0	66.5	**52.0**	61.9	**61.5**	62.5	**58.0**	64.1
	UniAct	79.0	69.8	61.5	**50.5**	**57.8**	59.0	61.5	50.5	**50.0**
	π_0	96.7	97.5	94.9	**76.9**	92.5	88.5	**80.9**	**88.0**	91.2
LIBERO Long	OpenVLA	51.7	48.5	32.8	**30.5**	36.0	30.5	**23.5**	36.0	**30.0**
	UniAct	46.5	32.5	28.0	**21.8**	**25.3**	**25.3**	30.3	18.8	**15.2**
	π_0	88.5	84.6	78.9	**64.4**	78.9	79.8	**73.0**	79.3	**76.9**
	MoDE	95.5	94.0	91.8	**80.3**	87.5	85.0	**84.3**	90.9	-
Habitat 2.0	LLARP	98.3	97.5	90.8	**60.7**	89.8	57.8	**46.2**	**83.7**	97.4

Table 2: Success rates of VLA models on different task suits and language perturbations. Bold type indicates the largest drop in the success rate across each group of perturbations.

Crowdsourcing participants instruction

Thank you for taking part in the experiment!

Imagine that you want to assign a task to a physical **robot**. The original task is written in text in the "**Original Text**" field.

You are required to **rephrase** this text and write it in your own words in the "**Your rephrasing**" (in your own words) field, keeping the original meaning.

Figure 2: The instruction that was shown to workers during crowdsourcing.

4 Experimental Results and Analysis

4.1 Command Paraphrasing by Human

According to the column "Human" in Table 2, natural command paraphrases lead to a lower number of successful episodes. Workers tend to use different synonyms in language commands, the vocabulary used is larger, and people do not tend to stick to any pattern of language command construction. Natural noise entering language commands tended to be a few words long and often contained various words related to politeness such as, but not limited to, "please" and "could".

In most cases, the success rate is reduced by 20%. However, in the case of UniAct model on LIBERO-Object tasks, the quality dropped by half.

The LLARP model appears to be fairly robust with human paraphrases, probably due to training on more complex and variant commands. We also conducted experiments with paraphrases by the DeepSeek V3 model and a template similar to the one used for the crowdsourcing platform. The greatest difference compared to human paraphrases amounts to 14% and is observed for the LLARP model, which addresses tasks involving navigation. In this case, human paraphrases exhibit greater variability in describing the location and the action to be performed with objects.

4.2 Irrelevant Context

All models showed performance degradation after adding irrelevant context. For a context with the same length as "Short", the largest drop in most cases is observed if the noise is semantically and lexically similar to a relevant command from the training set, i.e. belongs to the second group of contexts. On these types of contexts, at best a 10% drop can be observed, but more often models lose more than 50% of their quality.

An example of how context leads to dysfunctional robot behavior is shown in Figure 3. The target command is specified as 'find a lid and move it the black table', while the noise 'On the sofa there's an apple' is taken from a set of contexts "Location". Pointing to the location of an irrelevant object on the sofa triggers the robot to search for a target object on the sofa. In the absence of an

Start scene: navigate to sofa | Scene 2: pick lid, pick box, navigate to left counter | Scene 3: pick lego, pick strawberry, navigate to brown table | Scene 4: pick toy airplane, navigate to black table | Final scene: pick spoon

Figure 3: Demonstration of invalid robot behavior in a Habitat 2.0 simulator under the influence of irrelevant context "On the sofa there's an apple" for the target command "find a lid and move it the black table". The images correspond to the sequence of scenes from the episode. The captions under the scene images correspond to the actions that the robot executes.

object in the specified location, the robot starts to perform chaotic actions, trying to pick up various non-target objects while moving randomly around the scene.

As the context length increases, the performance of the model starts to decrease consistently for all considered cases. When the context size is equal to the length of the target command, the quality drop for contexts from the first group becomes comparable to the drop on semantically close context types; in some cases, may even surpass it.

This strong sensitivity of VLA models to linguistic distortions motivates the use of a multi-agent approach with command pre-processing. We examine a lightweight LLM-based pre-processing stage in a few-shot setting (see the Appendix B.2 for details). The results show that this form of filtering is effective mainly against simple random context, whereas more complex context types with semantic and lexical similarity to the command remain challenging.

4.3 Analysis

To better understand the mechanisms underlying the observed dependence of robustness on the type of irrelevant context, we conducted the following analysis. We considered the Llama 2 7B model used in the OpenVLA and LLARP models as the backbone, and extracted embedding representations separately for all contexts and target commands from the LIBERO benchmark at the final LLM layers. We then computed two variants of cosine similarity:

1. Cosine similarity between the mean embeddings of the last tokens for each context type and target commands for each task suite;
2. Cosine similarity between the mean embeddings over all tokens for each context type and target commands target commands for each task suite.

The resulting cosine similarities were compared against the number of successful episodes for each suite. Across both experiments, we observe a consistent qualitative trend. Therefore, we report here the first variant as this representation is more decisive for action generation (the remaining variant is provided in the Appendix C). As shown in Figure 4, the cosine similarity between context and target-command embeddings is directly associated with model performance. Higher similarity increases the relative contribution of the context through attention during command processing and shifts the final representation used for action generation away from the one on which the action head was trained. When the cosine similarity exceeds 0.75, the reduction in the number of successful episodes ranges from 20% to 50% depending on the task suite, with an average decrease of approximately 40%. Overall, this yields a clear negative correlation and suggests potential avenues for adversarial attacks on robots when the LLM or VLM backbone used within a VLA model is known.

4.4 Discussion

A qualitative analysis of rollout videos reveals characteristic behaviors of the manipulator in simulation for commands issued with and without additional context. On the LIBERO benchmark for commands without context, OpenVLA and π_0 generally attempt to grasp the correct object, and failures more often arise in the second phase of the task, where the agent must manipulate and trans-

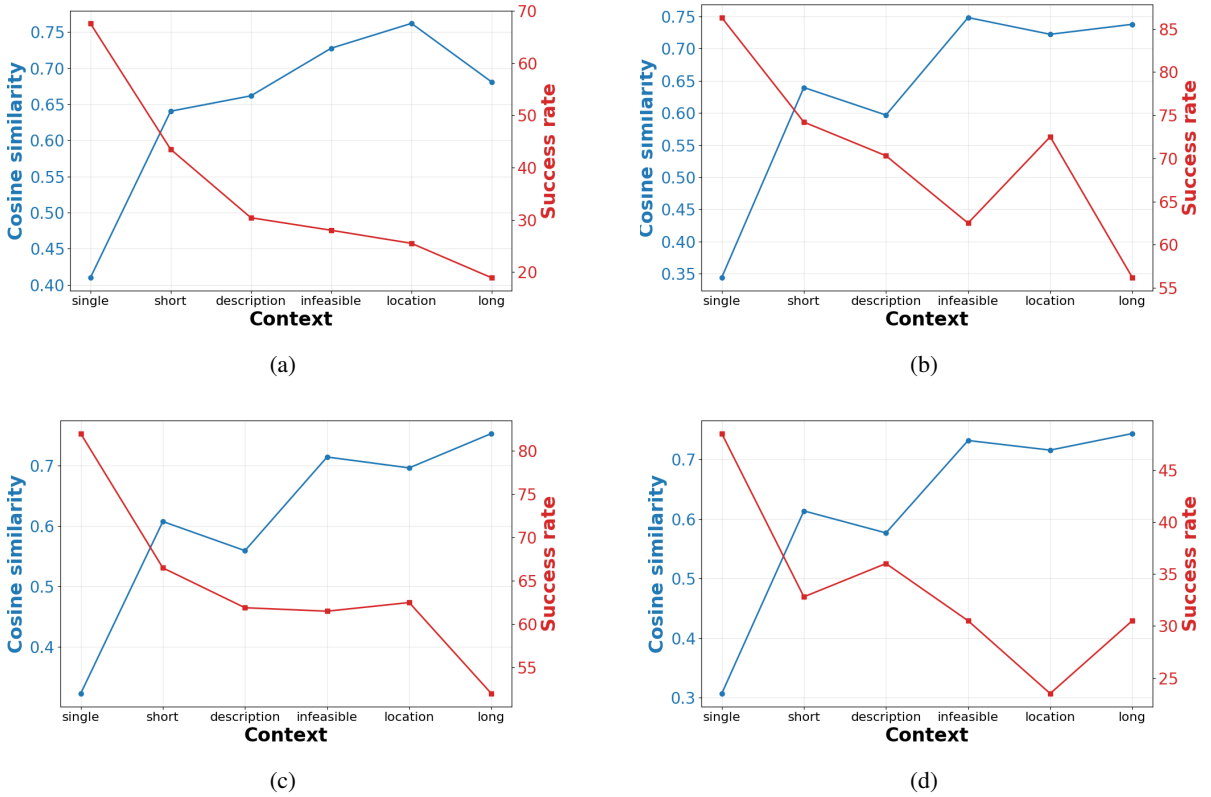


Figure 4: Inverse correlation between the success rate and the cosine similarity between context embeddings and target command embeddings on the LIBERO benchmark, aggregated by task suite: (a) `LIBERO-Goal`, (b) `LIBERO-Object`, (c) `LIBERO-Spatial`, and (d) `LIBERO-Long`.

port the correctly grasped target to its final location. When context is added, however, the manipulator may start interacting with incorrect objects already at the beginning of the episode. For example, for the command "turn on the stove" under a *"Long"* type context, the π_0 model picks up a cup instead of moving the gripper toward the stove knob. This behavior was not observed for the same command without context.

For the LLARP model in the Habitat simulator, we observe additional effects driven by the navigation subtask. Objects mentioned in lexically and semantically similar context are often implicitly present in the scene, which directly biases navigation: the model is drawn toward an irrelevant location mentioned in a *"Location"* type context rather than the true goal position.

Analysis based on cosine similarity further highlights that, as context becomes longer and more similar to the target command, attention to tokens corresponding to the target object (and its location) systematically decreases. This reduction in focus on task-critical words constitutes a non-trivial obstacle to reliable robot behavior under realistic human-robot communication conditions.

5 Conclusion

This study has thoroughly investigated the vulnerability of current vision-language-action models to human paraphrases and the presence of irrelevant linguistic context in robot manipulation commands. Experiments have shown that even minor textual noise can drastically reduce task success rates, with models showing pronounced sensitivity to certain types of irrelevant context. This behavior generalizes across VLA models based on different LLMs and is observed across various benchmarks and simulators. Using LLMs as filters for pre-processing and denoising instructions is effective for improving robustness and recovering performance in the presence of simple, random irrelevant context. However, semantically and lexically similar contexts remain challenging. Evaluating human-generated paraphrases further underscores the current limitations in the robustness of VLA models, which have primarily been trained and tested using synthetic data. Overall, this work highlights the critical importance of addressing linguistic variability to develop practical and widely utilized embodied AI systems.

Limitations

We have considered several reasonable groups of irrelevant context, but leave aside target commands with conditions and reasoning tasks, as these have been separately investigated in other works. We also set aside linguistic perturbations in the form of irrelevant characters and typos, as our primary focus is on the fundamental issues that may arise in humanrobot interaction and on the potential for a new class of adversarial attacks, rather than on random errors and misspellings that can be handled via preprocessing.

Ethics

Our work introduces a novel irrelevant context generation method to evaluate its impact on VLA robotic models. We acknowledge that our method for generating irrelevant linguistic context might be exploited to deliberately confuse deployed VLA systems. Nevertheless, we are convinced that the scientific value of openly documenting these vulnerabilities outweighs that misuse risk. By shedding light on VLA models' failures, we aim to catalyze safer and reliable embodied agents, and will release all code and data under a research-only license to promote responsible use.

Our study involves using crowdsourcing with paid participants to collect paraphrases of embodied AI commands created by humans. We paid assessors at rates above the average wage to ensure fair compensation for their time. This approach reflects our commitment to work ethics and respects the value of human contributions to AI research.

Crowdsourcing We used Toloka.ai as a crowdsourcing vendor. According to the user agreement and privacy policy, personal data typically includes information that can identify an individual, such as name, contact information, and other personal identifiers. Human paraphrases do not fall under this category. Moreover, we provide fully anonymized data that can not be linked to the people who wrote each text. Toloka policy allows for the sharing of anonymized data with third parties.

Acknowledgements

The work of Elena Tutubalina was supported within the framework of the HSE University Basic Research Program. We acknowledge the computational resources of the HPC facilities at HSE University.

References

Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. 2025. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1933–1943. IEEE.

Suneel Belkhale and Dorsa Sadigh. 2024. Minivla: A better vla with a smaller footprint.

Lucas Beyer*, Andreas Steiner*, André Susano Pinto*, Alexander Kolesnikov*, Xiao Wang*, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, and 16 others. 2024. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, and 1 others. 2024. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, and 32 others. 2023. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, and 273 others. 2023. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, and 1 others. 2023. Palm-e: An embodied multimodal language model.

Daniel P Jeong, Zachary Chase Lipton, and Pradeep Kumar Ravikumar. 2025. LLM-select: Feature selection with large language models. *Transactions on Machine Learning Research*.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. 2023. Vima: General robot manipulation with multimodal

prompts. In *Fortieth International Conference on Machine Learning*.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2025. Openvla: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *Preprint*, arXiv:2408.03326.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. 2024b. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, qiang liu, Yuke Zhu, and Peter Stone. 2023. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Artem Lykov, Valerii Serpiva, Muhammad Haris Khan, Oleg Sautenkov, Artyom Myshlyaev, Grik Tadevosyan, Yasheerah Yaqoot, and Dzmitry Tsetserukou. 2025. Cognitivedrone: A vla model and evaluation benchmark for real-time cognitive task solving and reasoning in uavs. *arXiv preprint arXiv:2503.01378*.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. 2023. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. 2024. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. 2024. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands.

Amit Parekh, Nikolas Vitsakis, Alessandro Suglia, and Ioannis Konstas. 2024. Investigating the role of instruction variety and task difficulty in robotic manipulation tasks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19389–19424, Miami, Florida, USA. Association for Computational Linguistics.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and 1 others. 2025. Spatialvla: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems*.

Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. 2025. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. In *The Thirteenth International Conference on Learning Representations*.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, and 1 others. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347.

Valerii Serpiva, Artem Lykov, Artyom Myshlyaev, Muhammad Haris Khan, Ali Alridha Abdulkarim, Oleg Sautenkov, and Dzmitry Tsetserukou. 2025. Racevla: Vla-based racing drone navigation with human-like behaviour. *arXiv preprint arXiv:2503.02572*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John M Turner, Noah D Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimír Vondruš, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel X Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, and 2 others. 2021. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems*.

Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. 2024. Large language models as generalizable policies for embodied tasks. In *The Twelfth International Conference on Learning Representations*.

Seyed Amin Tabatabaei, Sarah Fancher, Michael Parsons, and Arian Askari. 2025. Can large language models serve as effective classifiers for hierarchical

multi-label classification of scientific documents at industrial scale? In *Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Taowen Wang, Chen Han, James Chenhao Liang, Wenhao Yang, Dongfang Liu, Luna Xinyu Zhang, Qifan Wang, Jiebo Luo, and Ruixiang Tang. 2024a. Exploring the adversarial vulnerabilities of vision-language-action models in robotics. *arXiv preprint arXiv:2411.13587*.

Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. 2024b. Ladev: A language-driven testing and evaluation platform for vision-language-action models in robotic manipulation. *arXiv preprint arXiv:2410.05191*.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, and 1 others. 2025. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*.

Hongyin Zhang, Shuo Zhang, Junxi Jin, Qixin Zeng, Runze Li, and Donglin Wang. 2025. Robustvla: Robustness-aware reinforcement post-training for vision-language-action models. *arXiv preprint arXiv:2511.01331*.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. 2025. Universal actions for enhanced embodied foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C Knoll. 2025. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, and 35 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR.

A Appendix

In order to complete all the evaluations, we spent 1300 GPU hours utilizing 5 NVIDIA Tesla A100 GPUs.

A.1 Examples of Commands with Irrelevant Context

This subsection provides concrete examples of noisy commands used to evaluate the impact of irrelevant context on VLA models across different simulation environments (Habitat 2.0 and LIBERO benchmarks). The commands in Table 3–6 illustrate the insertion of various types of irrelevant context around the original robot commands to test model robustness.

These tables highlight the diversity and complexity of noise introduced to test model vulnerability.

Table 7 and 9 show the differences in the effect of noise inserted before and after the command.

B Irrelevant Context Filtering

B.1 Proposed Framework

Sec. 4 shows that the presence of irrelevant context leads to undesirable robot behavior. It is essential to extract the main command from the noisy text. Retraining the VLA model is a computationally and data-intensive process, which does not guarantee improved robustness of the resulting model. Since we consider different types of context, including a complex type in terms of semantic and lexical similarity, it can hardly be processed with templates. Therefore, we address this problem with LLMs, which have been recognized as powerful tools for selective classification, even in zero-shot settings (Jeong et al., 2025; Tabatabaei et al., 2025).

We investigate how models of varying sizes: tiny (FlanT5 Base, Qwen 2.5 0.5B Instruct), small (Qwen 2.5 1.5B Instruct, Llama 3.2 1B Instruct), medium (Qwen 2.5 3B Instruct, Llama 3.2 3B Instruct), and standard (MetaLlama38BInstruct)perform on a filtering task in a few-shot setting. We prompt models with the instruction, which contains three examples of context filtering. Different types of context are used, namely *"Short"*, *"Location"* and *"Infeasible"*. This prompt is specific and can improve filtering in more complex cases of irrelevant context. We also examined the instruction with only one context type *"Short"* in the examples. However, it performed poorly on semantically similar contexts (see Table 9 in Appendix).

Filtering instructions were adapted for the LLARP model and models for LIBERO benchmarks (see examples in Appendix Figure 7).

B.2 Instructions examples and results of the filtering framework

This subsection presents the prompt instructions for the first and second types filtering for both Habitat 2.0 (LLARP model) and the LIBERO benchmark.

Each prompt from Figure 7 includes three examples of filtering out short, location and infeasible types of irrelevant phrases that do not refer to scene objects or commands.

Each prompt from Figure 8 includes three examples of filtering out short irrelevant phrases that do not refer to scene objects or commands, i.e. context of the type *"Short"*. This type of context does not contain information about the training data. It allows to assess how generalizable a given filtering method is to other types of context. However, we found that this type of prompt demonstrates poor performance when filtering semantically similar contexts (Table 9), therefore all results in the main sections are presented for the second type of instruction.

This approach relies on few-shot prompting with LLMs, demonstrating its ability to discard irrelevant context effectively without knowledge of the robot's training process or task feasibility.

Table 9 demonstrates how this type of prompt instructions can generalize filtering across different types of context. As can be seen from the table, the generalization is generally present, but "Infeasible" type of noise requires additional information or examples about the robot's abilities.

Table 10 highlights the potential pitfalls for filtering framework, when important details can be accidentally removed.

B.3 Evaluation of Filtering Framework

The filter behaves differently on noisy commands for the LIBERO benchmark versus the LLARP model, due to differences in the underlying target commands. While LIBERO uses template-style commands (Appendix Table 5), LLARP was trained on more natural language with the navigation part (Appendix Table 3). As a result, on noisy LIBERO commands the filter does not change the target command regardless of whether it succeeds

Type	Command
Original	Find a sponge and move it to the right counter. Find a can and move it to the brown table. Find a strawberry and move it to the black table. Find a screwdriver and move it to the right counter. On the sink I need you to put a lego. Find a lemon and move it to the sofa. Find a bowl and move it to the right counter. Find a can and move it to the left counter.
Single	Although, find a mug on the right counter and move it to the tv stand. Meanwhile, find a wrench and move it to the left counter. Furthermore, find a orange and move it to the right counter. Consequently, find a lid and move it to the black table. Additionally, find a lid on the left counter and move it to the black table. Thus, find a wrench on the right counter and move it to the left counter. Nevertheless, find a mug and move it to the brown table. Whereas, find a toy airplane and move it to the right counter.
Short	Joyful moment watching birds. Find a orange and move it to the brown table. Anxiety peaks near deadlines. Find a toy airplane and move it to the right counter. Laughing while watching movie. Find a orange and move it to the brown table. Relieved after finishing chores. Find a toy airplane on the sink and move it to the black table. Excited gaming with friends. Find a wrench on the brown table and move it to the right counter. Embarrassed dancing in pajamas. Find a orange and move it to the right counter. Find a wrench and move it to the right counter. Inspired while painting walls. Regretful about skipped cleaning. Find a toy airplane and move it to the black table.
Long	He felt motivated cleaning the pantry and organizing everything, so find an orange and move it to the sink. He felt nostalgic holding his childhood paint once again. Find a mug on the right counter and move it to the TV stand. A burst of joy came with finishing her knitting project. Find a wrench and move it to the left counter. Late at night, anxiety crept in with pending homework reminders. Find a orange and move it to the right counter. She felt silly dancing barefoot to her favorite songs. Find a lid and move it to the black table. Inspiration struck while staring at the blank white canvas. Find a lid on the left counter and move it to the black table. Find a wrench on the right counter and move it to the left counter. He was satisfied after prepping meals for the entire week. Find a mug and move it to the brown table. The gloomy weather matched her tired and melancholy mood today.

Table 3: Examples of context inserted into commands for the LLARP model in the Habitat 2.0 simulator.

Type	Command
Location	Theres an apple on the TV stand, but find an orange and move it to the sink. On the sofa there's a lemon, but find a mug on the right counter and move it to the TV stand. On the sofa there's an apple, but instead find a wrench and move it to the left counter. Find a orange and move it to the right counter. I left my lid on the black table. Theres a wrench in the sink, but find a lid and move it to the black table. On the brown table there's an lego, but instead find a lid on the left counter and move it to the black table. I left my toy airplane on the brown table. Find a wrench on the right counter and move it to the left counter. I left my lid on the black table. Find a mug and move it to the brown table. Theres an orange on the right counter. Find a wrench on the right counter and move it to the sink. Theres a lego on the sofa. Find a plum and move it to the sofa. I left my apple on the left counter.
Description	Cup is a container for liquids. Find an orange and move it to the sink. Clamp is useful for woodworking. Find a mug on the right counter and move it to the TV stand. Lemon is citrus. Find a lid and move it to the sink. Orange is juicy fruit. Find a wrench on the left counter and move it to the right counter. Plum is a fruit. Find a mug and move it to the sofa. Lego has colorful blocks. Bring the wrench from the right counter to the left counter. Bring the bowl from the TV stand to the right counter? Lid protects from spilling. Find a toy airplane on the right counter and move it to the left counter. Strawberry is delicious. Find a mug and move it to the sink. Bowl is for serving food. Find a mug and move it to the black table. Ball is a round object.
Infeasible	Bake a pie with peach slices. Find an orange and move it to the sink. Cut ribbons with sharp scissors. Find a mug and move it to the sofa. Spray cleanser on the countertops. Bring the toy airplane from the right counter to the left counter. Bite into the soft plum slowly. Find a mug and move it to the sofa. Garnish tea with fresh lemon slices. Find a lid and move it to the sink. Bounce a small ball around. Bring the wrench from the sofa to the TV stand. Clamp the pipe for repair. Find a lid and move it to the left counter. Find a orange and move it to the right counter. Kick the colorful ball outside. Bite into the soft plum slowly. Glide the toy airplane across the room. Bring the wrench from the sofa to the TV stand. Explore new stories in your book.

Table 4: Examples of context inserted into commands for the LLARP model in the Habitat 2.0 simulator.

in detecting the context or not. For LLARP, removing context can lead to paraphrasing.

Figure 5 illustrates how the number of successes varies for LLARP in the Habitat 2.0 simulator depending on filtering by LLMs with sizes up to 3B. As can be seen from the figure, even small filters with up to 0.5B parameters handle filtering of random context well. However, as semantic similarity increases, the quality of filtering decreases and becomes comparable to the results before filtering for Flat-T5 Base for the context type *"Location"*, while Llama 3.2 3B Instruct demonstrates the maximum gain up to 79% of successful episodes, which is still lower then the original quality.

For LIBERO commands, starting at the 3B model size, only Llama 3.2 is able to recover the majority of the original commands (see Figure 6). It should be noted, that Llama 3.2 1B Instruct had difficulty following the template in the instruction,

Type	Command
Original	turn on the stove put the bowl on top of the cabinet put the wine bottle on the rack pick up the cream cheese and place it in the basket pick up the milk and place it in the basket pick up the chocolate pudding and place it in the basket pick up the black bowl between the plate and the ramekin and place it on the plate pick up the black bowl next to the ramekin and place it on the plate put both the cream cheese box and the butter in the basket put both moka pots on the stove
Single	however open the middle drawer of the cabinet therefore put the bowl on the stove moreover put the wine bottle on top of the cabinet open the top drawer and put the bowl inside although meanwhile put the bowl on top of the cabinet furthermore push the plate to the front of the stove consequently put the cream cheese in the bowl turn on the stove thus put the bowl on the plate additionally nevertheless put the wine bottle on the rack
Short	open the middle drawer of the cabinet feeling lonely in bed morning coffee feels serene put the bowl on the stove restless pacing after sunset put the wine bottle on top of the cabinet nostalgia strikes after dinner open the top drawer and put the bowl inside put the bowl on top of the cabinet confused thoughts while cooking push the plate to the front of the stove quiet satisfaction reading books focusing deeply at night put the cream cheese in the bowl stress builds during arguments turn on the stove calm mornings with tea put the bowl on the plate peaceful napping on weekends put the wine bottle on the rack
Long	she found peace sipping tea in the afternoon light open the middle drawer of the cabinet frustration loomed as he searched for his misplaced keys put the bowl on the stove quiet happiness filled the room while rearranging photo albums put the wine bottle on top of the cabinet he felt nostalgic holding his childhood toy once again open the top drawer and put the bowl inside a burst of joy came with finishing her knitting project put the bowl on top of the cabinet push the plate to the front of the stove late at night anxiety crept in with pending homework reminders she felt silly dancing barefoot to her favorite songs put the cream cheese in the bowl turn on the stove inspiration struck while staring at the blank white canvas he was satisfied after prepping meals for the entire week put the bowl on the plate the gloomy weather matched her tired and melancholy mood today put the wine bottle on the rack

Table 5: Examples of context inserted into commands for models evaluated on the LIBERO benchmark.

and its results underwent minor post-processing, where the filtered command was extracted from the overall generated text that contained variants of the phrase filtered:, filter:, and etc.

If we further increase the model size and examine the Meta-Llama-3-8B-Instruct model, the detailed analysis shows the following. In the case of the LIBERO template, almost all types of irrel-

Type	Command
Location	open the middle drawer of the cabinet the bowl is in the stove the mug is in the microwave put the bowl on the stove put the wine bottle on top of the cabinet the moka pots are on the stove the cream cheese is in the basket open the top drawer and put the bowl inside the mug is on the plate put the bowl on top of the cabinet push the plate to the front of the stove the chocolate pudding is on the plate put the cream cheese in the bowl the mug is on the right plate turn on the stove the book is inside of the caddy put the bowl on the plate the wine bottle is in the basket the wine bottle in the cabinet pick up the milk and place it in the basket
Description	open the middle drawer of the cabinet ball is a round object put the bowl on the stove ball bounces on surfaces clamp holds objects together put the wine bottle on top of the cabinet clamp is useful for woodworking open the top drawer and put the bowl inside hammer is tool for driving nails put the bowl on top of the cabinet push the plate to the front of the stove hammer has a metal head put the cream cheese in the bowl screwdriver tightens screws screwdriver comes in various sizes turn on the stove put the bowl on the plate padlock secures items with a key padlock are made of metal put the wine bottle on the rack
Infeasible	toss the ball softly outdoors put both the alphabet soup and the tomato sauce in the basket put both the cream cheese box and the butter in the basket enjoy a juicy plum after lunch snack on a sweet pear tonight turn on the stove and put the moka pot on it a ripe peach feels refreshing put the black bowl in the bottom drawer of the cabinet and close it put the white mug on the left plate and put the yellow and white mug on the right plate bite into a crisp apple pick up the book and place it in the back compartment of the caddy eat fresh strawberries with yogurt solve the tricky rubiks cube put the white mug on the plate and put the chocolate pudding to the right of the plate read your favorite book tonight put both the alphabet soup and the cream cheese box in the basket put both moka pots on the stove sip tea from the cup quietly fly the toy airplane for fun put the yellow and white mug in the microwave and close it

Table 6: Examples of context inserted into commands for models evaluated on the LIBERO benchmark.

evant context were filtered out successfully (see Table 11), and the target command remained unchanged. The only exceptions were commands that were preceded by infeasible non-target commands of the type *"Infeasible"*. For the LLARP model and VLA models on LIBERO-Goal and LIBERO-Long benchmarks, the performance is recovered by more than 90%.

C Analysis of cosine similarity vs success rates

Figure 9 show the dependencies of success rate on cosine similarity between the mean embeddings of all tokens for each context type and target commands separately for task suites from the LIBERO benchmark.

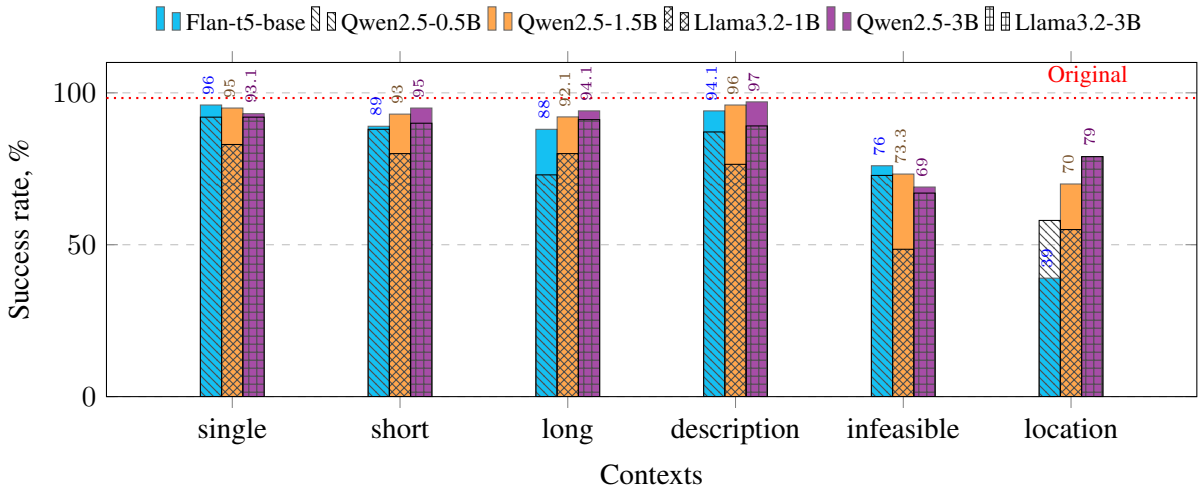


Figure 5: Success rates for LLARP in the Habitat 2.0 simulator for commands with different types of irrelevant context after filtering by LLMs of various sizes using a few-shot prompt.

Setup		Original	Linking	Short	Long	Location	Description	Infeasible
Goal	Noise Before	77.5	71.5	48.0	20.5	29.0	35.5	29.0
	Noise After	77.5	63.7	39.0	17.3	22.0	25.3	27.0
Object	Noise Before	87.3	86.3	77.7	53.0	74.0	72.3	64.0
	Noise After	87.3	86.3	70.7	59.3	71.0	68.3	61.0
Spatial	Noise Before	85.3	84.3	75.0	56.0	67.0	73.7	64.0
	Noise After	85.3	79.7	58.0	48.0	58.0	50.0	59.0
Long	Noise Before	51.7	51.3	35.3	31.7	25.0	40.3	32.0
	Noise After	51.7	45.7	30.3	29.3	22.0	31.7	29.0

Table 7: Success rate of the OpenVLA model on the LIBERO-Goal, Object, Spatial and Long task suits depending on irrelevant context, color-coded by value magnitude.

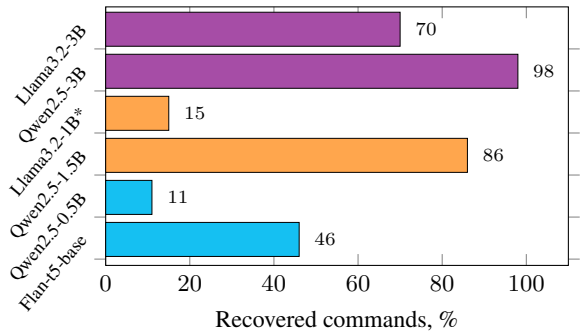


Figure 6: Ratio of recovered commands from the LIBERO benchmark averaged across task suites and all types of irrelevant context

Habitat 2.0 (LLARP) Instruction

Convert the text into a command for the robot by removing all irrelevant context. Save details that may be important for the robot to find and move objects according to the target command. Here are some examples:
Text: Today is a nice weather. Pick an apple from the TV stand and move it to the left counter.
Filtered: Pick an apple from the TV stand and move it to the left counter.

Text: There's an apple in the fridge, but set down a strawberry and a lemon on the right counter.
Filtered: Set down a strawberry and a lemon on the right counter.

Text: Tighten screws with a screwdriver. Find a mug on the right counter and move it to the TV stand.
Filtered: Find a mug on the right counter and move it to the TV stand.

Now, convert the following text:
Text: instruction

LIBERO Instruction

Convert the text into a command for the robot by removing all irrelevant context. Save details that may be important for the robot to find and move objects according to the target command. Here are some examples:
Text: today is a nice weather open the middle drawer of the cabinet
Filtered: open the middle drawer of the cabinet

Text: the mug is on the right plate pick up the black bowl between the plate and the ramekin and place it on the plate
Filtered: pick up the black bowl between the plate and the ramekin and place it on the plate

Text: spray the counter with cleanser pick up the cream cheese and place it in the bowl
Filtered: pick up the cream cheese and place it in the bowl

Now, convert the following text:
Text: instruction

Figure 7: Examples of instructions with 3 different types of irrelevant context used in the filtering framework.

Setup		Original	Linking	Short	Long	Location	Description	Infeasible
Goal	Noise Before	91.5	91.3	82.8	49.8	60.0	75.8	69.0
	Noise After	91.5	92.0	73.0	39.8	51.3	61.8	50.3
Object	Noise Before	97.5	97.3	94.5	83.8	85.0	92.5	92.5
	Noise After	97.5	97.5	95.0	86.0	86.8	91.0	92.5
Spatial	Noise Before	96.75	97.0	94.0	77.0	82.0	92.0	90.5
	Noise After	96.75	97.0	94.5	79.8	81.5	93.0	88.8
Long	Noise Before	88.5	85.0	79.0	66.5	72.5	80.0	84.5
	Noise After	88.5	85.3	82.5	66.0	74.0	80.5	83.8

Table 8: Success rate of the π_0 model on the LIBERO-Goal, Object, Spatial and Long task suits depending on irrelevant context, color-coded by value magnitude.

Habitat 2.0 (LLARP) Instruction

Convert the text into a command for the robot by removing all irrelevant context. Save details that may be important for the robot to find and move objects according to the target command. Here are some examples:
Text: Today is a nice weather. Pick an apple from the TV stand and move it to the left counter.
Filtered: Pick an apple from the TV stand and move it to the left counter.

Text: I think its a good idea to set down an strawberry and a lemon on the right counter.
Filtered: Set down a strawberry and a lemon on the right counter.

Text: The fridge door is open because of my mistake. Find a mug on the right counter and move it to the TV stand.
Filtered: Find a mug on the right counter and move it to the TV stand.

Now, convert the following text:
Text: instruction

LIBERO Instruction

Convert the text into a command for the robot by removing all irrelevant context. Save details that may be important for the robot to find and move objects according to the target command. Here are some examples:
Text: today is a nice weather open the middle drawer of the cabinet
Filtered: open the middle drawer of the cabinet

Text: i think its a good idea to push the plate to the front of the stove
Filtered: push the plate to the front of the stove

Text: it is quiet at home put the cream cheese in the bowl
Filtered: put the cream cheese in the bowl

Now, convert the following text:
Text: instruction

Figure 8: Examples of instructions with 1 type of irrelevant context used in the filtering framework.

Setup	Original	Linking	Short	Long	Location	Description	Infeasible
Noise Before	98.3	97.5	90.8	60.7	46.2	89.8	57.8
Noise After	98.3	97.3	93.1	62.7	58.9	75.9	65.0
Filter 3-shots 1-type	98.3	98.3	98.3	98.3	98.3	96.0	77.6
Filter 3-shots 3-types	98.3	98.3	98.3	98.3	98.3	95.7	94.9

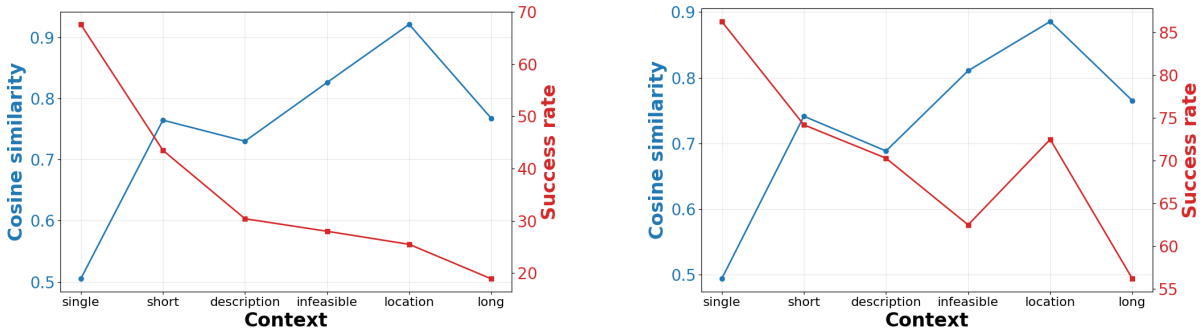
Table 9: Success rate of the LLARP model in the Habitat 2.0 simulator depending on irrelevant context, color-coded by value magnitude.

Before filtering	After filtering
Scissors has sharp blades. Find a lid on the right counter and bring it to the sofa.	Bring it to the sofa.
Spoon scoops liquids. Find the wrench in the sink and bring it to the sofa.	Bring the wrench to the sofa.
Peach is for desserts. Find the wrench on the sofa and bring it to the TV stand.	Bring the wrench to the TV stand.
Spatula scrapes food. Find the wrench on the sofa and bring it to the brown table.	Bring the wrench to the brown table.
turn on the stove and put the moka pot on it	put the moka pot on it

Table 10: All examples of filtering a noisy command while removing important details of the target command.

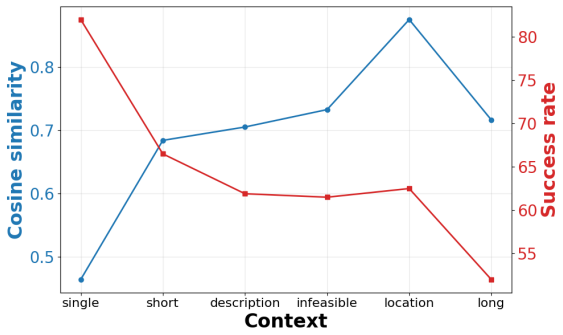
Environment	Model	Original	Single	Short	Long	Location	Description	Infeasible	Human
LIBERO Goal	OpenVLA + F	77.5	77.5	77.5	77.5	77.5	77.5	73.0↓	59.2↑
	UniAct + F	67.5	67.5	67.5	67.5	67.5	67.5	66.0↓	44.2↑
LIBERO Object	OpenVLA + F	87.3	87.3	87.3	87.3	87.3	87.3	87.3	79.1↓
	UniAct + F	86.5	86.5	86.5	86.5	86.5	86.5	86.5	45.6↑
LIBERO Spatial	OpenVLA + F	85.3	85.3	85.3	85.3	85.3	85.3	85.3	55.0↓
	UniAct + F	79.0	79.0	79.0	79.0	79.0	79.0	79.0	48.0↓
LIBERO Long	OpenVLA + F	51.0↓	51.7	51.7	51.7	51.7	51.7	46.7↓	35.0↓
	UniAct + F	46.5	46.5	46.5	46.5	46.5	46.5	37.5↓	19.6↑
	MoDE + F	95.5	95.5	95.5	95.5	95.5	95.5	93.5↓	-
Habitat 2.0	LLARP + F	98.3	98.3	98.3	98.3	98.3	95.7↓	94.9↓	82.1↓

Table 11: Success rates of models on the LIBERO benchmark and Habitat 2.0 simulator on commands after filtering with MetaLlama38BInstruct. Arrows correspond to cases where original commands were not fully recovered.

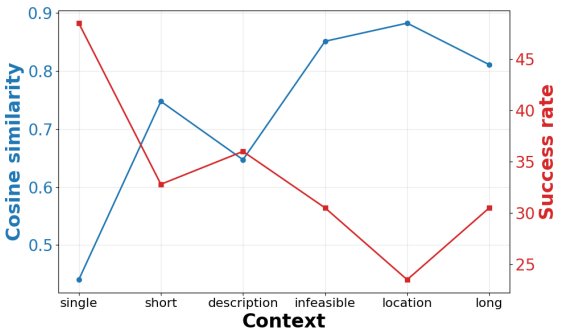


(a)

(b)



(c)



(d)

Figure 9: Inverse correlation between the success rate and the cosine similarity between the mean context embeddings and target command embeddings from the last transformer layer on the LIBERO benchmark, aggregated by task suite: (a) `LIBERO-Goal`, (b) `LIBERO-Object`, (c) `LIBERO-Spatial`, and (d) `LIBERO-Long`.