VLA-HALLUCINATION: DIAGNOSING AND MITIGATING HALLUCINATION FAILURES IN VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models have recently become an important technique for general-purpose robotic manipulation. However, existing VLA models still exhibit clear limitations in generalization to unseen tasks. We find that VLA task failures can be categorized into manipulation failures and hallucination failures. The latter refers to cases where the model-generated actions are inconsistent with the target task or the surrounding environment, and constitutes a major factor limiting VLA generalization. Further analysis reveals that an important cause of hallucination failures is that task-relevant visual-semantic information preserved in the upstream VLM is not effectively utilized during downstream action generation, causing the generated actions to deviate from the current task semantics. To address this issue, we propose **Action-to-Visual Semantic Alignment (AVSA)**. AVSA introduces a lightweight visual-semantic reconstructor that reconstructs visual-semantic information conditioned on the action, online measures the alignment between Action Expert outputs and VLM visual semantics, and uses this signal to correct actions during Flow Matching, thereby aligning action generation with the upstream visual-semantic understanding. AVSA can be integrated with multiple mainstream VLA models without retraining the VLA backbone, and consistently reduces hallucination rates while improving task success rates across diverse unseen settings, including environment changes, semantic changes, and paraphrased instructions. For example, on LIBERO-Para, AVSA improves the task success rate of $\pi_{0.5}$ by 7.1%. Overall, our results demonstrate that explicitly diagnosing and aligning action generation with visual semantics provides an effective approach to mitigating hallucination failures and improving the generalizability of VLA models.

1 INTRODUCTION

With language understanding and visual perception capabilities, VLA models are expected to adapt robot actions to new environments and novel language instructions, enabling robots to complete unseen tasks (Kim et al., 2024). However, compared with Vision-Language Models (VLMs), VLA models have not yet demonstrated the same level of generalization ability. The success rates of VLA models drop significantly when evaluated on unseen scenes or unseen instructions (Fei et al., 2025; Zhou et al., 2025; Li). We argue that hallucination is a key factor limiting their generalization.

In this work, VLA hallucination is defined as cases where the actions generated by a VLA model are inconsistent with the current visual scene or language instruction. To characterize this phenomenon, task failures are categorized into hallucination failures and manipulation failures. As shown in Figure 1, we define task failures caused by deviations from task-related information as hallucination failures, such as following visual cues to execute an unintended task while ignoring the instruction. In contrast, a robot task may still fail even when the VLA model correctly understands the relevant visual and semantic information. We refer to these cases as manipulation failures, which may result from limitations in fine-grained manipulation, action stability, spatial understanding, or action decoding (Li et al., 2025; Pan et al., 2026). Therefore, the two types of failures can be distinguished according to their underlying causes. In real-world environments, we analyze video trajectories with a video understanding model to determine whether the current target task can be inferred from the ex-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

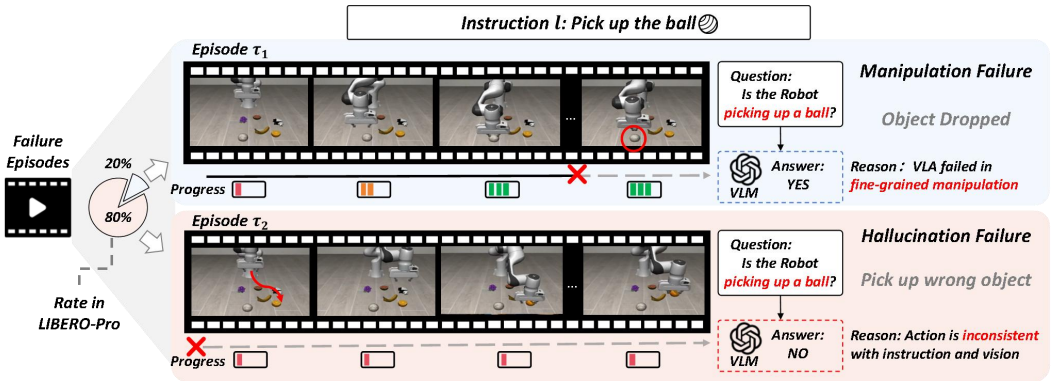


Figure 1: **Hallucination Failures and Manipulation failure.** The figure shows how failure types are distinguished by analyzing failure trajectories with a video understanding model. The top row shows grasping failures caused by execution errors, where the generated actions remain consistent with target task. In contrast, the bottom row shows hallucination failures, where the generated actions deviate from the visual-semantic information.

ecuted behavior, thereby distinguishing between the two failure types. In simulation environments, we instead use privileged physical information provided by the simulator for failure attribution.

To quantify this phenomenon, we propose the VLA hallucination rate (HR), which measures the proportion of hallucination failures among all failures. As shown in Figure 2, VLA models exhibit only limited failures on seen tasks on LIBERO (Liu et al., 2023), where failures are mainly attributed to manipulation failures. However, when evaluated on unseen tasks, the failure rate (FR) increases substantially, and hallucination failures become a dominant failure mode. For instance, $\pi_{0.5}$ (Intelligence et al., 2025) shows no hallucination failures on the LIBERO, while its HR rises sharply on LIBERO-Para Kim et al. (2026) and LIBERO-Pro (Zhou et al., 2025), and is especially high on the Pos and Task subsets of LIBERO-Pro, corresponding to target-object position perturbations and task-semantic variations. Similar trends are observed for SmolVLA (Shukor et al., 2025) and XVLA (Zheng et al., 2025), demonstrating that VLA models are substantially more prone to hallucination on unseen tasks.

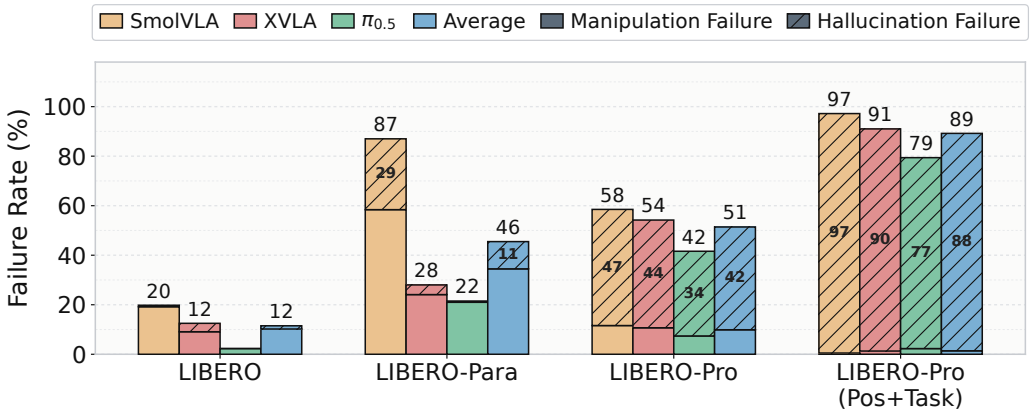
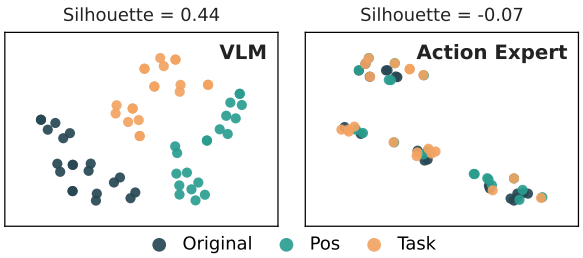


Figure 2: FR and HR across LIBERO, LIBERO-Para, LIBERO-Pro, and LIBERO-Pro (Pos+Task).

To further investigate the source of hallucination failures, we analyze how task-relevant visual-semantic information is preserved along the pathway from the VLM to action generation Wang et al. (2026); Xu et al. (2026b); Yin & Zhang (2026). Specifically, during hallucination trajectories, we extract the VLM representation obtained after vision-language interaction, which encodes task-relevant visual semantics, and denote it as H^I. We also extract the corresponding action representation H^A from the Action Expert. For the same base tasks, we construct controlled position and task-semantic variants and measure the category separability of H^I and H^A using the Silhouette Score. As shown in Figure 3, H^I consistently exhibits higher separability across all three VLA mod-



Features	SmolVLA	XVLA	$\pi_{0.5}$
VLM	0.11	0.27	0.44
Action Expert	−0.04	−0.07	−0.07

Figure 3: Feature separability under hallucination failures. We analyze the feature separability of VLM and Action Expert representations under hallucination on the original task and two task variants. Left: For $\pi_{0.5}$, VLM representations remain relatively separable across task variants, whereas Action Expert representations exhibit clear mixing. Right: Across SmolVLA, XVLA, and $\pi_{0.5}$, the Silhouette Scores of VLM representations are higher than those of Action Expert representations.

els, with scores of 0.11, 0.27, and 0.44 for SmolVLA, XVLA, and $\pi_{0.5}$, respectively. In contrast, the corresponding H^A yields negative Silhouette Scores, indicating that the Action Expert representations fail to effectively distinguish among different task variants. This degradation in separability from H^I to H^A suggests that task-relevant visual-semantic information can remain preserved in the upstream VLM but deteriorate along the pathway toward action generation, revealing an important source of VLA hallucination.

Existing VLA models primarily follow a one-way process from visual-semantic understanding to action generation, without explicitly verifying whether the generated actions deviate from the task objective. Consequently, these deviations cannot be detected or corrected during generation, which may ultimately lead to hallucination failures. Therefore, motivated by enabling the Action Expert to better leverage the visual-semantic information provided by the VLM, we propose AVSA (Action-to-Visual Semantic Alignment) to mitigate such deviations. When generalizing to unseen scenes and instructions, the generated actions may deviate from the visual and semantic information encoded by the upstream VLM. AVSA explicitly monitors such deviations and aligns the actions with the visual-semantic information through action correction. During inference, AVSA introduces a visual-semantic reconstructor that reconstructs the visual semantics implied by the current action and online measures its deviation from the reference visual-semantic information. When a deviation is detected, AVSA further corrects the action, thereby constraining the generated action toward the visual semantics.

This mechanism of continuously monitoring whether generated actions deviate from the task objective and correcting such deviations in time enables AVSA to effectively mitigate hallucination failures and improve VLA generalization to unseen tasks. For example, under the Task perturbation of LIBERO-Pro, AVSA improves the success rate of $\pi_{0.5}$ from 19.2% to 25.5% while reducing the hallucination rate from 97.1% to 84.8%. Similar improvements are also observed on LIBERO-Para, RoboTwin, and real-world robotic tasks.

The main contributions of this paper are summarized as follows:

- **We propose a task failure classification method for VLA hallucination.** By analyzing failure trajectories, we categorize VLA failures into manipulation failures and hallucination failures, and further introduce the hallucination rate to quantify the prevalence and impact of hallucination failures.
- **To mitigate VLA hallucination, we propose AVSA, a hallucination mitigation method.** AVSA alleviates hallucination failures by online detecting the alignment between actions and visual semantics and correcting deviant actions.
- **Experiments on mainstream VLA models demonstrate that our method consistently reduces hallucination rates and improves generalization on unseen tasks.** Specifically, across different VLA models, our method reduces the hallucination rate by an average of 11.5% on LIBERO-Para and by 8.5% on LIBERO-Pro, while also improving task success rates in both simulation and real-world environments.

2 PRELIMINARIES

For flow-matching-based VLA models, given a visual observation I and a language instruction q, the upstream VLM first jointly encodes the visual and language inputs and produces a middle visual representation H^I together with a language representation H^q. These representations are subsequently passed to the downstream Action Expert as condition for action generation.

At the t-th Flow Matching step, the Action Expert predicts a velocity field from the current intermediate action state A_t under this conditioning:

$$v_t = f_\theta \left(A_t, t; H^I, H^q \right), \tag{1}$$

and progressively updates the action state along the predicted velocity field until the final action chunk A is obtained.

Since H^I is produced after vision-language interaction in the VLM, it encodes both the visual context of the current observation and task-related semantics conditioned on the language instruction. As demonstrated by the representation analysis in Figure 3, H^I remains discriminative across controlled visual and semantic task variants even in hallucination trajectories, whereas the corresponding Action Expert representation H^A exhibits substantially degraded separability. This observation motivates us to use H^I as the reference **visual-semantic representation** and explicitly measure whether the semantics encoded by the generated action remain aligned with it.

3 METHOD

AVSA (Action-to-Visual Semantic Alignment) is designed to mitigate hallucination caused by generated actions deviating from the current visual-semantics. Within each VLA cycle of reasoning, action generation, and execution, the model does not know whether the action produced by the downstream Action Expert is aligned with the upstream VLM's visual-semantic understanding. Therefore, after action generation, we online analyze this alignment and intervene in the action to maximize consistency with visual semantics. To this end, we introduce a lightweight visual-semantic reconstructor that reconstructs visual-semantic information conditioned on the action. We present AVSA from four aspects: (1) problem formulation; (2) training of the visual-semantic reconstructor; (3) online alignment monitoring during inference; and (4) gradient-guided action alignment.

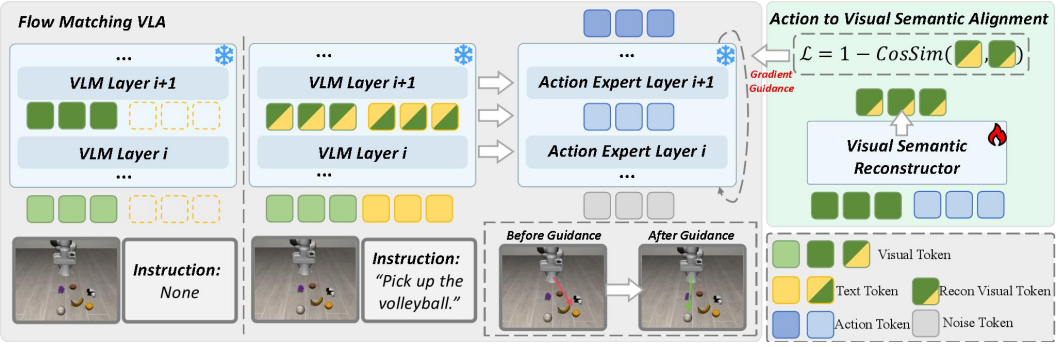


Figure 4: Overview of AVSA. After action generation, AVSA uses a visual-semantic reconstructor to assess alignment between the downstream action and the upstream VLM visual semantics, and applies gradient-guided intervention when deviation is detected.

3.1 PROBLEM FORMULATION

Following Sec. 2, we use the task-conditioned visual representation H^I produced by the VLM as the reference visual semantics of the current task. Our goal is to determine whether a candidate action chunk A generated by the Action Expert is semantically consistent with this upstream visual understanding.

To isolate the task-specific semantics encoded in the action, we additionally obtain a task-agnostic visual representation $H^{I,\emptyset}$ by feeding the same observation I to the VLM with an empty instruction. For a candidate action chunk A, we extract its corresponding Action Expert representation H^A.

We hypothesize that if A is semantically consistent with the current task, then its representation H^A, together with the task-agnostic scene context $H^{I,\emptyset}$, should retain sufficient task-related information to recover the target visual semantics H^I. Based on this assumption, we introduce a visual-semantic reconstructor R_ϕ that predicts the action-implied visual semantics from $(H^{I,\emptyset}, H^A)$, and define the action–visual semantic consistency score as

$$S(A) = \mathrm{CosSim}\left(R_\phi(H^{I,\emptyset}, H^A), H^I\right). \tag{2}$$

A higher $S(A)$ indicates that the candidate action is more consistent with the VLM's task-conditioned visual understanding, whereas a lower score suggests semantic deviation.

3.2 TRAINING THE VISUAL SEMANTIC RECONSTRUCTOR

The visual-semantic reconstructor is trained to recover task-conditioned visual semantics from visual context and the corresponding action representation. During this stage, the pretrained VLA policy (VLM and Action Expert) remains frozen. Given a training sample of observation, instruction, and action chunk, we compute the task-agnostic and task-conditioned visual features together with the Action Expert representation, and minimize

$$\mathcal{L}_{\mathrm{recon}} = \left\| R_\phi(\mathbf{H}^{I,\emptyset}, \mathbf{H}^A) - \mathbf{H}^I \right\|_2^2. \tag{3}$$

This objective requires R_ϕ to infer what task-related visual semantics an action implicitly expresses, rather than merely judging whether the action resembles training-set actions. To avoid language shortcuts, we restrict direct language information from entering the action representation used by R_ϕ, so that prediction relies mainly on the semantics encoded by the action itself.

3.3 ONLINE ACTION-VISUAL SEMANTIC MONITORING

At inference time, AVSA first builds the semantic references of the current task (task-agnostic and task-conditioned visual features). These references stay fixed while the current action chunk is generated, so they can be computed once and reused throughout Flow Matching. At generation step t, the Action Expert predicts a velocity field for the noisy intermediate action state. Because the intermediate state is still noisy, we first reconstruct the clean action implied by the current trajectory, extract its Action Expert representation, and evaluate the online alignment score S_t with the same cosine-similarity form as $S(\mathbf{A})$. This provides an internal feedback signal *before* execution, unlike environment feedback that is available only after interaction. Additionally, correction is activated only when S_t falls below a threshold γ; otherwise we keep the original VLA trajectory to avoid unnecessary intervention.

3.4 GRADIENT-GUIDED ACTION ALIGNMENT

The online alignment score introduced in Sec. 3.3 provides a measure of whether the action implied by the current Flow Matching trajectory is consistent with the task-conditioned visual semantics. When the alignment score falls below the threshold γ, we regard the current action trajectory as exhibiting a potential semantic deviation and use the alignment score as an inference-time guidance signal to correct the action generation process.

Specifically, at Flow Matching step t, we define the alignment objective as

$$\mathcal{L}_{\mathrm{align}}^t = 1 - S_t, \tag{4}$$

where S_t measures the consistency between the visual semantics implied by the current action and the task-conditioned visual representation produced by the VLM. Minimizing $\mathcal{L}_{\mathrm{align}}^t$ therefore encourages the generated action to encode visual semantics that are more consistent with the upstream visual-semantic understanding.

Table 1: Results on LIBERO-Pro across perturbation types.

Model	Ori		Obj		Pos		Sem		Task	
	SR	HR	SR	HR	SR	HR	SR	HR	SR	HR
SmolVLA	80.3	4.8	80.3	9.6	0.5	99.5	71.8	9.2	5.0	99.5
SmolVLA+AVSA	**81.0**	**3.2**	**83.0**	**6.4**	**9.0**	**88.0**	**80.5**	**6.1**	**12.0**	**90.5**
XVLA	87.5	4.5	87.3	9.4	1.5	98.6	77.8	8.7	16.5	98.6
XVLA+AVSA	**88.5**	**3.1**	**87.5**	**7.8**	**6.3**	**89.5**	**78.5**	**5.9**	**17.8**	**96.5**
$\pi_{0.5}$	97.7	0	97.4	8.5	22.0	97.1	95.1	9.1	19.2	97.1
$\pi_{0.5}$+AVSA	**98.5**	**0**	**97.6**	**6.1**	**24.3**	**91.0**	**95.4**	**6.9**	**25.5**	**84.8**

Rather than directly modifying the predicted action, we apply the alignment guidance to the velocity field predicted by the Action Expert. Since the velocity field determines the evolution of the intermediate action state during Flow Matching, adjusting the velocity provides a natural way to steer the subsequent generation trajectory. We back-propagate the alignment objective through the action representation and the visual-semantic reconstructor to obtain its gradient with respect to the current velocity field. The corrected velocity is given by

$$\tilde{v}_t = v_t - \eta \nabla_{v_t} \mathcal{L}^t_{\text{align}}, \tag{5}$$

where η denotes the guidance strength.

The gradient $\nabla_{v_t} \mathcal{L}^t_{\text{align}}$ indicates the direction in the velocity space that increases the discrepancy between the action-implied semantics and the reference visual semantics. Moving in the opposite direction therefore steers the action trajectory toward a region with higher action–visual semantic consistency. The corrected velocity $\tilde{v}_t$ is subsequently used in the standard Flow Matching update, allowing the original action generation process to be preserved while introducing visual-semantic guidance only when necessary.

4 Experiments

We introduce a task failure analysis mechanism and the HR metric to characterize different failure cases after task execution. And we evaluate the effectiveness of the proposed AVSA on LIBERO, RoboTwin, and a real-world AgileX PiPER robot using representative VLA models, including π0.5, SmolVLA, and XVLA.

Our experiments are conducted from the following three perspectives:

- **Evaluation of success rate and hallucination rate:** We investigate whether the proposed method can improve the generalization ability of VLA models while reducing the hallucination rate.
- **Mechanism analysis:** We analyze the accuracy of AVSA in reconstructing visual semantics under hallucination, the effectiveness of gradient guidance in improving visual semantic alignment, and a case study showing how AVSA corrects a hallucination-failure trajectory into a successful one.
- **Ablation studies:** We study the impact of different hyperparameter settings on VLA performance.

4.1 Main Experimental Results

Evaluation on LIBERO. As shown in Tables 1 and 2, we evaluate AVSA under various perturbation settings on LIBERO-Pro and LIBERO-Para, respectively. On LIBERO-Pro, AVSA consistently improves the performance of different VLA models across perturbation types including Ori, Obj, Pos,

Table 2: Results on LIBERO-Para.

	SmolVLA	SmolVLA+AVSA	XVLA	XVLA+AVSA	$\pi_{0.5}$	$\pi_{0.5}$+AVSA
SR	78.5	**81.2**	72.0	**78.4**	13.0	**20.1**
HR	2.3	**1.8**	13.8	**5.7**	32.9	**23.5**

Table 3: Results on a subset of RoboTwin 2.0 (SR / HR, %).

Simulation Task	$\pi_{0.5}$		$\pi_{0.5}$+AVSA		**XVLA**		**XVLA+AVSA**	
	SR	HR	SR	HR	SR	HR	SR	HR
Adjust Bottle	92.0	25.0	**93.0**	**14.3**	84.0	12.5	**86.0**	**7.1**
Beat Block Hammer	48.0	13.5	**50.0**	**10.0**	40.0	16.7	**42.0**	**13.8**
Blocks Ranking RGB	22.0	20.5	**24.0**	**18.4**	15.0	22.4	**17.0**	**19.3**
Blocks Ranking Size	10.0	23.3	**12.0**	**20.5**	8.0	23.9	**10.0**	**21.1**
Click Alarmclock	68.0	9.4	**70.0**	**6.7**	58.0	11.9	**60.0**	**7.5**
Click Bell	50.0	14.0	**52.0**	**10.4**	40.0	16.7	**42.0**	**13.8**
Dump Bin Bigbin	86.0	14.3	**87.0**	**7.7**	75.0	8.0	**77.0**	**4.3**
Grab Roller	96.0	50.0	**97.0**	**33.3**	90.0	20.0	**91.0**	**11.1**
Handover Block	50.0	14.0	**52.0**	**10.4**	40.0	16.7	**42.0**	**13.8**
...								
Move Pillbottle Pad	24.0	19.7	**26.0**	**17.6**	18.0	22.0	**20.0**	**18.8**
Average (in %)	54.6	20.4	**56.3**	**14.9**	46.8	17.1	**48.7**	**13.1**

Sem, and Task. For example, under Task perturbation, the success rate of $\pi_{0.5}$ increases from 19.2% to 25.5%, while the hallucination rate decreases from 97.1% to 84.8%. On LIBERO-Para, AVSA similarly improves success rates and reduces hallucination rates under diverse perturbations, indicating that the method enhances the robustness of VLA models to task-semantic changes, language paraphrasing, and environmental variations.

Evaluation on RoboTwin. As shown in Table 3, we further evaluate AVSA on a subset of the RoboTwin 2.0 benchmark and report SR and HR. We report results for $\pi_{0.5}$ and XVLA, as well as their counterparts with AVSA.

Evaluation in the real-world environment. Real-world setup. We evaluate on the AgileX PiPER 6-DoF robotic arm. The evaluation task set consists of two subsets: an instruction perturbation task set and a vision perturbation task set to verify the effectiveness of mitigating hallucination. Each subset contains three unseen tasks, and each task is evaluated over 20 trials. Figure 5 shows one task and its two task variants.

Real-world experimental results. Table 4 shows that our method improves the success rate by an average of 11.7% and reduces the hallucination rate by 14.4%, demonstrating its transferability to the real world.

Table 4: Results on real-world tasks.

Model	**Instruction**		**Vision**		**Avg**	
	SR	**HR**	**SR**	**HR**	**SR**	**HR**
$\pi_{0.5}$	23.3	73.9	53.3	46.4	38.3	60.2
$\pi_{0.5}$+AVSA	**38.3**	**56.8**	**61.7**	**34.8**	**50.0**	**45.8**

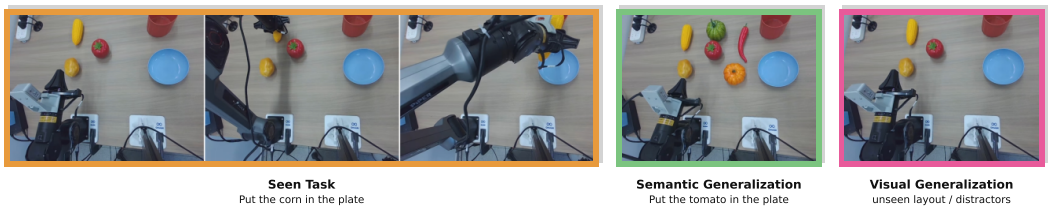


Figure 5: Demo of our real-world experimental setup. Left (orange): seen task *Put the corn in the plate*. Middle/right: unseen variants for instruction perturbation (*Put the tomato in the bowl*) and vision perturbation (seen task with distractors).

4.2 Mechanism Analysis

We analyze the working mechanism of AVSA from three aspects: (1) the accuracy of visual semantics reconstructed by the Visual Semantic Reconstructor; (2) how gradient guidance improves visual-semantic alignment when the reconstructed semantics deviate from the ground truth; and (3) a case study of converting a hallucination-failure trajectory into a successful one.

Visual Semantic Reconstructor Accuracy. On success and manipulation-failure trajectories, the Visual Semantic Reconstructor recovers visual semantics that are highly aligned with the ground-truth target; on hallucination-failure trajectories, the alignment is low. As shown in Figure 6 (Left), the alignment degrees of success and manipulation-failure trajectories are mainly concentrated above 0.7, whereas those of hallucination-failure trajectories are primarily below 0.5, indicating that the Visual Semantic Reconstructor can effectively distinguish hallucination failures from other trajectories. Figure 6 (Right) further visualizes the original H^I and reconstructed $\hat{H}^I$ under hallucination failures: before correction, the two representations form clearly separated distributions, whereas after AVSA correction, the reconstructed representations shift toward and substantially overlap with the original visual-semantic representations.

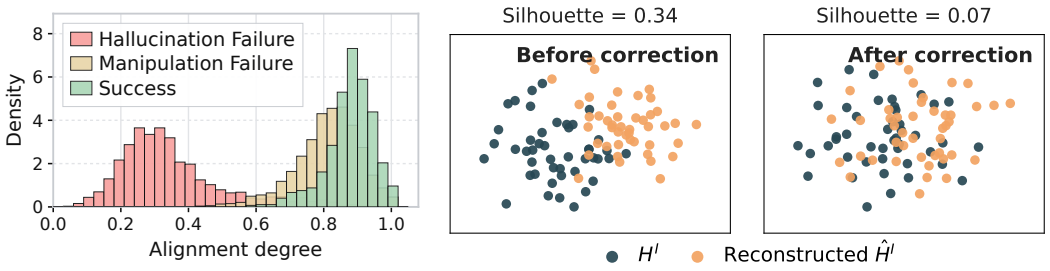


Figure 6: Visual-semantic reconstruction under hallucination. Left: density of alignment degree for Success and two Failure types. Right: t-SNE of the original H^I and the reconstructed $\hat{H}^I$ before and after AVSA correction.

Gradient Guidance Effect. To evaluate the effect of gradient guidance on visual-semantic alignment, we select the original tasks together with several representative task variants, including language paraphrases, target-object position changes, and task-semantic variations. We then measure the **average visual-semantic alignment degree** with and without gradient guidance. As shown in Figure 7, AVSA improves visual-semantic alignment under Ori, Para, Pos, and Task settings. On Ori, the average alignment gain is about 0.08; on Para, Pos, and Task, the gains are about 0.14, 0.18, and 0.22. This indicates that AVSA retains a clear generalization advantage under harder perturbations.

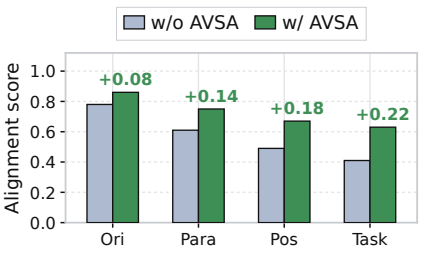


Figure 7: Effect of gradient guidance on visual-semantic alignment under Ori, Para, Pos, and Task.

Solving Hallucination Failure Case. Figure 8 shows that at t=3 the trajectories diverge; AVSA raises $S(A)$ from 0.54 to 0.75 and resolves the hallucination.

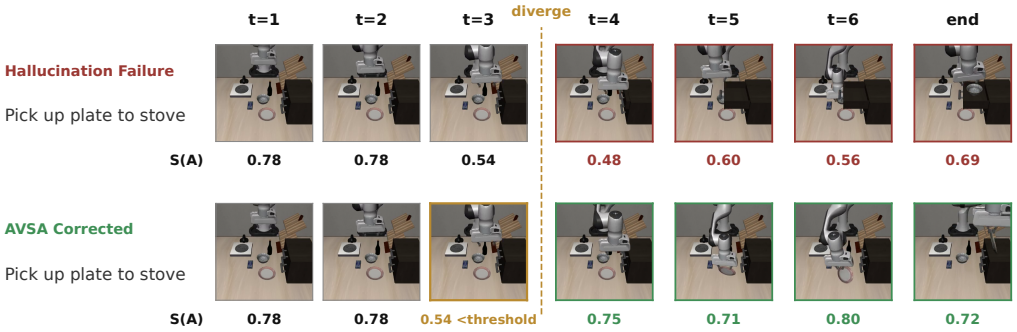


Figure 8: Case study of resolving a hallucination failure. Frames t=1–3 share the same $S(A)$; at t=3, $S(A)$ falls below the threshold and AVSA corrects the bottom trajectory.

4.3 Ablation Studies

We conduct ablation studies to analyze two design choices of AVSA: (1) the guidance strength η used for gradient-guided action alignment, and (2) the input design of the visual-semantic reconstructor.

Table 5: Effect of the guidance strength η. Success rate (SR).

	η=0.5	η=1.0	η=1.5	η=2.0	η=2.5
SR ↑	51.8	54.3	**56.8**	54.1	53.5

Table 6: Ablation on the input design of the visual-semantic reconstructor. Success rate (SR).

	H^A	$H^{I,\emptyset}+H^A$	$H^{I,\emptyset}+H^A+H^q$
SR ↑	50.6	**56.8**	51.2

Effect of Guidance Strength. The guidance strength determines the magnitude of intervention that AVSA applies to the original VLA trajectory. As shown in Table 5, when η is too small, the correction signal is weak and cannot effectively redirect actions that have already deviated from the target visual semantics. Conversely, when η is too large, it may over-correct the velocity field and interfere with the original action-generation trajectory.

Effect of Visual-Semantic Reconstructor Inputs. We further study the input design of the visual-semantic reconstructor by comparing three variants: (i) H^A only; (ii) $H^{I,\emptyset}+H^A$, our default design; and (iii) $H^{I,\emptyset}+H^A+H^q$. As shown in Table 6, using H^A alone removes the visual context of the current scene, making it more difficult to recover task-relevant visual semantics. Introducing $H^{I,\emptyset}$ provides task-agnostic scene information and allows the reconstructor to better associate the action with relevant objects and spatial relations. We also evaluate the effect of explicitly providing the language representation H^q. Although H^q contains task information, it may introduce a language shortcut, allowing the reconstructor to recover target semantics directly from the instruction rather than from the semantics expressed by the action.

5 Conclusion

We investigate hallucination failures in VLA models and identify a key bottleneck in the vision-semantics–action pathway. We propose AVSA to mitigate these issues by online aligning generated actions with VLM visual semantics through a lightweight visual-semantic reconstructor and gradient-guided correction. Experiments on simulation and real-world tasks demonstrate that AVSA effectively reduces hallucination rates and improves generalization. Our work highlights the importance of aligning action generation with visual semantics for building reliable VLA systems.

AI USE STATEMENT

In this work, we used generative AI tools to draft and edit parts of the manuscript for readability, to suggest paper structure and wording, to assist with literature search and summarization, to help format references and LaTeX tables, and to create or modify scientific figures and plotting scripts. We have not used generative AI tools to generate synthetic datasets for our main experiments, to invent core mathematical claims, or to produce unverified experimental results. All research ideas, method design, experimental protocols, quantitative results, and final scientific claims were proposed, verified, and approved by the authors. We reviewed all AI-assisted text, code, and figures for correctness and originality, and we take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies hallucination failures of Vision-Language-Action models in simulation and on a physical robotic arm. Experiments involve only standard robot manipulation benchmarks and controlled tabletop tasks with toy objects; no human subjects or personally identifying data are collected. Real-robot evaluations are conducted in a laboratory setting with standard safety practices for collaborative manipulators. We do not release harmful dual-use capabilities beyond improving the reliability of existing VLA systems, and we encourage careful deployment when transferring such methods to unstructured real-world environments.

REPRODUCIBILITY STATEMENT

We provide implementation and evaluation details needed to reproduce our results in the main paper and appendix, including the AVSA architecture and training configuration (Appendix B), hallucination detection protocol and metrics (Appendix C.1), preprocessing (Appendix C.2), and real-world task definitions (Appendix D). Experiments are built on publicly available VLA backbones and standard benchmarks (LIBERO / LIBERO-Pro / LIBERO-Para, RoboTwin, and our described real-world protocol). Upon acceptance, we plan to release code and configuration files to facilitate reproduction of the reported results.

REFERENCES

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

Junzhe Chen, Tianshu Zhang, Shiyu Huang, Yuwei Niu, Linfeng Zhang, Lijie Wen, and Xuming Hu. Ict: Image-object cross-level trusted intervention for mitigating object hallucination in large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4209–4221, 2025.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in VLAs. *arXiv preprint arXiv:2602.17659*, 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Chanyoung Kim, Minwoo Kim, Minseok Kang, Hyunwoo Kim, and Dahuin Jung. Libero-para: A diagnostic benchmark and metrics for paraphrase robustness in vla models. *arXiv preprint arXiv:2603.28301*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

Quanyi Li. Task reconstruction and extrapolation for $\pi 0$ using text latent, 2025. *URL https://arxiv.org/abs/2505.03500*.

Zhuohao Li, Yinghao Li, Jian-Jian Jiang, Lang Zhou, Tianyu Zhang, Jiadong Yin, Mu Lin, Yi-Lin Wei, and Wei-Shi Zheng. ReViP: Mitigating false completion in vision-language-action models with vision-proprioception rebalance. *arXiv preprint arXiv:2601.16667*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Zeting Liu, Zida Yang, Zeyu Zhang, and Hao Tang. Evovla: Self-evolving vision-language-action model. *arXiv preprint arXiv:2511.16166*, 2025.

Yi Pan, Miao Pan, Qi Lu, Jiaming Huang, Man Zhang, Siteng Huang, Xin Li, Jie Zhang, Yongliang Shen, Xuhong Zhang, et al. Vla-corrector: Lightweight detect-and-correct inference for adaptive action horizon. *arXiv preprint arXiv:2607.01804*, 2026.

Carlota Parés-Morlans, Nils Kuhn, Isabel Liu, Alberta Longhini, and Jeannette Bohg. Demystifying when and why VLAs fail in contact-rich tasks and how to fix them. *arXiv preprint arXiv:2608.01402*, 2026.

Ye Qiu, Zhaoxin Fan, Qingchen Yu, Faguo Wu, Hongwei Zheng, Yifan Sun, and Wenjun Wu. Entropy-optimized contrastive decoding for hallucination suppression in vision-language-action models. *Neurocomputing*, 669:132507, 2026.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Shengyu Si, Yuanzhuo Lu, Ruimeng Yang, Ziyi Ye, Zuxuan Wu, and Yu-Gang Jiang. VLA-Pro: Cross-task procedural memory transfer for vision-language-action models. *arXiv preprint arXiv:2605.29562*, 2026.

Harold Soh and Eugene Lim. Action hallucination in generative visual-language-action models. *arXiv preprint arXiv:2602.06339*, 2026.

Wei Suo, Lijun Zhang, Mengyang Sun, Lin Yuanbo Wu, Peng Wang, and Yanning Zhang. Octopus: Alleviating hallucination via dynamic contrastive decoding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29904–29914, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Xuan Wang, Yinan Wu, Haoran Duan, and Jungong Han. Quovla: Quotient space for vision-language-action models. *arXiv preprint arXiv:2605.24890*, 2026.

Haiweng Xu, Sipeng Zheng, Hao Luo, Wanpeng Zhang, Ziheng Xi, and Zongqing Lu. Unmasking the illusion of embodied reasoning in vision-language-action models. *arXiv preprint arXiv:2604.18000*, 2026a.

Yuan Xu, Youheng Shi, Chengyang Li, Wentao Zhu, and Yizhou Wang. Semantic anchoring for robotic action representations. *arXiv preprint arXiv:2607.13597*, 2026b.

Zhaokai Yin and Zhipeng Zhang. Grounded semantic re-binding for robust instruction generalization in vision-language-action models. *arXiv preprint arXiv:2608.02497*, 2026.

Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Juntao Dai, Yuanpei Chen, and Yaodong Yang. Safevla: Towards safety alignment of vision-language-action model via constrained learning. *Advances in Neural Information Processing Systems*, 38:153335–153373, 2026a.

Ninghao Zhang, Bin Zhu, Shijie Zhou, and Jingjing Chen. Restoring linguistic grounding in VLA models via train-free attention recalibration. *arXiv preprint arXiv:2603.06001*, 2026b.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Zhengru Fang, Yi Nie, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026c.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

APPENDIX

A RELATED WORK

A.1 HALLUCINATION IN VLM

Hallucination in VLMs has been widely studied in the literature. In this context, hallucination mainly refers to the phenomenon where the textual content generated by the model is inconsistent with the visual image input (Bai et al., 2024). For example, a VLM may describe objects or contents that do not exist in the given image. ICT (Chen et al., 2025) mitigates hallucination by performing image-level and object-level dual attention intervention during inference, guiding the model to focus on both the overall image and fine-grained visual contents. Octopus (Suo et al., 2025) introduces a dynamic recognition module that adaptively selects the optimal contrastive decoding strategy at each step of text generation, thereby producing more accurate textual outputs.

A.2 HALLUCINATION IN VLA

Hallucination in VLA models differs from hallucination in VLMs and can lead to more severe consequences. VLA hallucination is an action-level error that can cause a robot to grasp the wrong object, perform irrelevant tasks, or even perform potentially dangerous behaviors (Zhang et al., 2026a).

The term VLA hallucination has been mentioned in several recent works. EvoVLA (Liu et al., 2025) introduces the concept of stage hallucination, where the model assumes that a task stage has been completed while it is in fact unfinished. One work regards hallucination as an illusion that VLA models appear to possess language and visual understanding capabilities, while their actual behavior may instead rely on data memorization and shortcut learning (Xu et al., 2026a). Another work defines VLA hallucination as the generation of robot actions or trajectories that violate physical laws or environmental constraints (Soh & Lim, 2026).

Beyond these definitions, a growing line of work studies concrete failure modes that partially overlap with our manipulation-versus-hallucination taxonomy. Fang et al. (2026) analyze *counterfactual failures*, where VLAs follow visual shortcuts instead of language intent, and propose Counterfactual Action Guidance as a training-free dual-branch inference remedy. Closely related, Zhang et al. (2026b) identify *linguistic blindness*, in which policies continue visually plausible execution under contradictory instructions, and introduce train-free attention recalibration to restore linguistic grounding. Li et al. (2026) study *false completion*, where proprioceptive progress dominates visual evidence and the policy prematurely treats a failed execution as finished; this failure is closer to our hallucination category when the action semantics diverge from the observed scene, while still

being distinct from contact-rich *manipulation* failures. Parés-Morlans et al. (2026) provide a complementary account of precision and force failures in contact-rich tasks, which primarily fall under manipulation failures in our taxonomy rather than instruction–scene semantic deviations.

Several inference-time methods also aim to suppress action-level hallucination or improve reliability of frozen VLAs. Qiu et al. (2026) propose Entropy-Optimized Contrastive Decoding to reduce action hallucinations without additional training. Zhang et al. (2026c) wrap a frozen VLA as a retryable primitive inside a memory-guided agentic harness, offering a systems-level alternative to correcting frozen policies under perturbed deployments. Si et al. (2026) instead transfer cross-task procedural memories via retrieved LoRA adapters at inference time, improving unseen-task generalization on RoboTwin and real robots without fully retraining the backbone. Kim et al. (2026) introduce LIBERO-Para, a diagnostic benchmark and metrics for paraphrase robustness in VLA models, which we use to evaluate whether AVSA mitigates paraphrase-induced failures. In contrast, AVSA targets a specific bottleneck along the vision-semantics–action pathway: it reconstructs action-implied visual semantics and applies gradient-guided correction when alignment with upstream VLM features degrades.

B Implementation Details

B.1 Model Architecture

AVSA is attached to a frozen pretrained VLA backbone and does not modify the VLM or Action Expert parameters. The backbone follows the standard VLM+Action Expert pipeline described in Sec. 2. The only trainable component is the visual-semantic reconstructor R_ϕ, which maps $(\mathbf{H}^{I,\emptyset}, \mathbf{H}^{A})$ to a reconstructed visual-semantic representation $\hat{\mathbf{H}}^{I}$ in the same feature space as $\mathbf{H}^{I}$. In our implementation, R_ϕ is a lightweight pre-norm Transformer with 12 learnable layer queries; its width matches the Action Expert hidden size of the host VLA.

B.2 Training Configuration

During reconstructor training, the pretrained VLA policy remains frozen. We sample observation–instruction–action tuples from the corresponding demonstration datasets of each backbone, compute $(\mathbf{H}^{I,\emptyset}, \mathbf{H}^{A}, \mathbf{H}^{I})$, and minimize the reconstruction loss in Sec. 3.2. We optimize R_ϕ with AdamW at a learning rate of $3{\times}10^{-4}$ for 5000 steps with batch size 64, and train on a single NVIDIA RTX 4090 GPU. We train a separate reconstructor for each backbone ($\pi_{0.5}$, SmolVLA, and XVLA) so that feature spaces remain consistent with the host model.

B.3 Hyperparameters

At inference time, AVSA is controlled mainly by two hyperparameters: (i) the alignment threshold γ, which determines when gradient guidance is activated; and (ii) the guidance strength η, which scales the velocity-field correction. Unless otherwise stated, we use the default setting selected on a held-out validation split and report sensitivity to η in Sec. 4.3. Performance peaks around $\eta{=}1.5$: smaller values under-correct drifted actions, while larger values over-perturb the Flow Matching velocity field.

C Dataset and Evaluation Details

We evaluate on three settings: (i) LIBERO and its generalization suites LIBERO-Pro (Zhou et al., 2025) / LIBERO-Para (Kim et al., 2026); (ii) a subset of RoboTwin 2.0 under Easy and Hard protocols; and (iii) real-world trials on an AgileX PiPER 6-DoF arm. For real-world evaluation, the task set contains an instruction-perturbation subset and a vision-perturbation subset; each subset has three unseen tasks, and each task is evaluated over 20 trials.

C.1 Details of Hallucination Detection

Following the main text, a failed trial is labeled as a *hallucination failure* if the executed actions are inconsistent with the current visual scene or language instruction (i.e., the behavior deviates from task-related visual-semantic information). It is labeled as a *manipulation failure* if the executed behavior remains consistent with the target task, but the trial still fails due to execution issues such as grasp slips, imprecise placement, or unstable contact.

Simulation. In LIBERO / LIBERO-Pro / LIBERO-Para and RoboTwin, we use privileged simulator state for failure attribution. Given a failed episode, we inspect object contacts, grasped-object identity, end-effector–object relations, and goal-related predicates exposed by the simulator. If these signals show that the robot is interacting with the instructed target object/goal in a task-consistent way, but the success predicate remains false, the trial is marked as a manipulation failure; otherwise it is marked as a hallucination failure.

Real world. In real-world trials, privileged state is unavailable. We therefore analyze the RGB video of each failed trajectory with QwenVL-7B (Wang et al., 2024). Given the language instruction and the rollout video, the model judges whether the target task can be inferred from the executed behavior. If the behavior remains consistent with the target task but the trial fails, we label a manipulation failure; if the behavior clearly diverges from the instruction or the relevant visual semantics, we label a hallucination failure. To validate this detector, we annotate 100 real-world trajectory videos and compare QwenVL-7B predictions with human labels; the detection success rate reaches 97.0%. Ambiguous cases in the main evaluation are reviewed by a human annotator.

Hallucination rate. Following the main text, the hallucination rate is defined as the proportion of hallucination failures among all failures,

$$\mathrm{HR} = \frac{N_{\text{hallucination failure}}}{N_{\text{failure}}}, \tag{6}$$

while the failure rate is $\mathrm{FR} = N_{\text{failure}}/N_{\text{executed}}$. Successful trials are excluded from both the numerator and the denominator of HR.

C.2 Preprocessing

Images are resized to the native resolution expected by each VLA backbone. Language instructions are tokenized with the backbone tokenizer without additional prompt engineering beyond the task text. For reconstructor training, empty-instruction features $\mathbf{H}^{I,\emptyset}$ are obtained by forwarding the same observation with an empty language input, as described in Sec. 3.1.

C.3 Evaluation Metrics

We report Success Rate (SR), Failure Rate (FR), and Hallucination Rate (HR). SR / FR / HR are computed as described in Appendix C.1. Alignment analyses additionally use cosine similarity $S(A)$ and Silhouette Score on representation clusters.

D Real World Details

D.1 Hardware Setup

We conduct real-world experiments on an AgileX PiPER 6-DoF robotic arm. RGB observations are captured from a fixed third-person camera overlooking the workspace and resized to the native resolution of the host VLA backbone (Appendix C.2). Inference runs with the same frozen backbone and AVSA reconstructor configuration as in simulation (Appendix B); only the embodiment and observation stream change.

D.2 Task Suite

The evaluation suite targets hallucination under distribution shift and contains two complementary subsets (Figure 5):

- **Instruction perturbation.** Held-out object–container pairings: *Put the corn in the plate*, *Put the potato in the plate*, and *Put the tomato in the bowl*.
- **Vision perturbation.** The seen instruction *Put the corn in the plate* is kept unchanged, while distractors are added to the workspace under three different configurations.

Table 7 lists the training skill and all evaluation tasks. Each evaluation subset contains three tasks; each task is run for 20 trials.

Table 7: Real-world training and evaluation tasks on AgileX PiPER. Instruction-perturbation tasks use held-out object–container pairings; vision-perturbation tasks reuse the seen instruction with added distractors.

Split	Subset	Language instruction
Training	Seen	*Put the corn in the plate*
Evaluation	Instruction	*Put the corn in the plate*
	Instruction	*Put the potato in the plate*
	Instruction	*Put the tomato in the bowl*
	Vision	*Put the corn in the plate* (+ distractors)
	Vision	*Put the corn in the plate* (+ distractors)
	Vision	*Put the corn in the plate* (+ distractors)

D.3 Evaluation Protocol

Each unseen task is evaluated for 20 independent trials, yielding 60 trials per subset and 120 trials in total for the reported $\pi_{0.5}$ comparison in Table 4. A trial is marked successful if the instructed goal is completed within the episode budget; otherwise it is a failure. Failed trials are further labeled as hallucination vs. manipulation failures following Appendix C.1 (QwenVL-7B video judgment with human review on ambiguous cases). We report SR and HR for the instruction and vision subsets, as well as their average.

D.4 Example Evaluation Tasks

Figure 5 illustrates representative evaluation settings in the main text. Below we show dual-view rollouts for the two perturbation types. For each example, we uniformly sample six frames from the full trajectory and display both the top and wrist cameras.

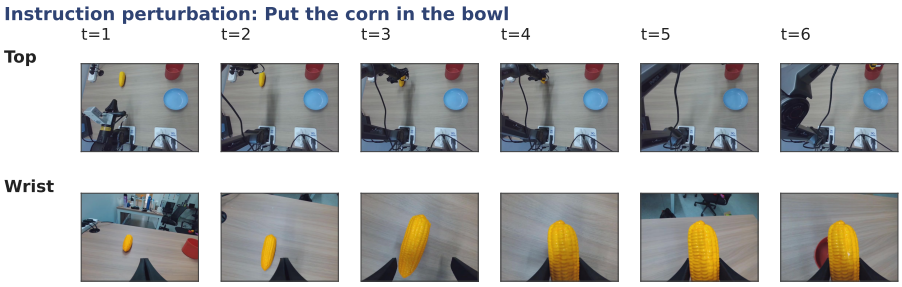


Figure 9: Instruction-perturbation example under *Put the corn in the bowl*.

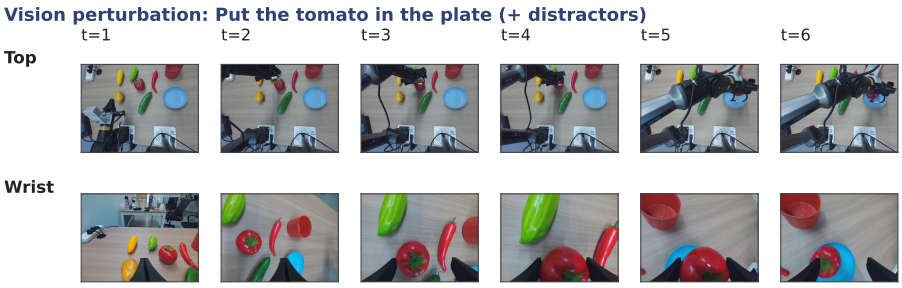


Figure 10: Vision-perturbation example under *Put the tomato in the plate* with distractors.