STAR: STEERING ACTION REPRESENTATIONS WITH PROPRIOCEPTION FOR VLA INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action models (VLAs) map language instructions and visual observations to robot actions, showing strong capabilities across robotic manipulation tasks. However, in a probing experiment, we find that action representations near the action output do not fully preserve task-relevant visual grounding. Motivated by this late-layer grounding gap, we introduce STAR, for **ST**eering **A**ction **R**epresentations, a training-free inference method for pre-trained VLAs. STAR applies an exponential moving average over late-layer residual updates, allowing action representations to inherit object-aware update directions from preceding layers within the mixing region without updating model weights or adding trainable parameters. It further uses recent proprioceptive states to adapt the mixing strength, using conservative mixing when execution dynamics change rapidly. Experiments on robot manipulation benchmarks show that STAR improves late-layer object awareness and increases average task success rates by 4.0 percentage points.

1 INTRODUCTION

Despite the tremendous success of vision-language models (VLMs) (Liu et al., 2023; Deitke et al., 2025; Karamcheti et al., 2024; Li et al., 2025b; Beyer et al., 2024; Bai et al., 2025), previous work has shown that they may rely too heavily on language priors, producing responses that are not fully grounded in the visual input (Li et al., 2023b; Leng et al., 2024; Huang et al., 2024; Favero et al., 2024; Wang et al., 2025). Meanwhile, recent studies also observe that vision-language-action models (VLAs) exhibit similar behavior patterns, such as insufficient use of task-relevant visual cues (Song et al., 2025; Pani & Yang, 2026; Vo et al., 2025; Yu et al., 2025; Li et al., 2024a) or distraction from irrelevant visual tokens (Li et al., 2026; Song et al., 2026; Zhou et al., 2026; Li et al., 2025a). Inspired by these works, we ask a key question: *how strongly are action representations in VLAs grounded in task-relevant visual evidence*?

To probe the dependence of action representations on visual evidence, we design a controlled experiment. As illustrated in Figure 1 (a), we mask the target object pixels by replacing them with the mean RGB values of the image. By masking the visual information important for action execution, we expect the action representations to differ from those produced without interference. To measure the representation change induced by object masking, we define an *object awareness* score, computed as $\|h^k_{\mathrm{m}} - h^k_{\mathrm{o}}\| \,/\, \|h^k_{\mathrm{o}}\|$. Here, h^k is the hidden state of the k-th layer in Figure 1(b), while o and m denote the original and masked observations, respectively. This metric provides a heuristic estimate of how strongly task-relevant visual evidence affects the action representation. A larger score indicates a larger difference in the representation after masking. The resulting object-awareness score for $\pi_{0.5}$ (Physical Intelligence et al., 2025) is reported in Figure 1 (c). We find that, during native inference, object awareness increases through early-to-middle layers and drops in late layers. This suggests that visual evidence is gradually aggregated into action representations from early to middle layers, but fades away in late layers. This probing experiment demonstrates a grounding gap in late layers during VLA inference.

To address the gap, we introduce **STAR**, for **ST**eering **A**ction **R**epresentations, a training-free method that regularizes the action representations in late layers without updating model weights. Intuitively, STAR steers late-layer action representations by carrying residual updates across the late-layer mixing region. Specifically, STAR applies an exponential moving average (EMA) (Holt, 2004) over late-layer updates. When execution dynamics change rapidly, such as during end-effector rotation

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.

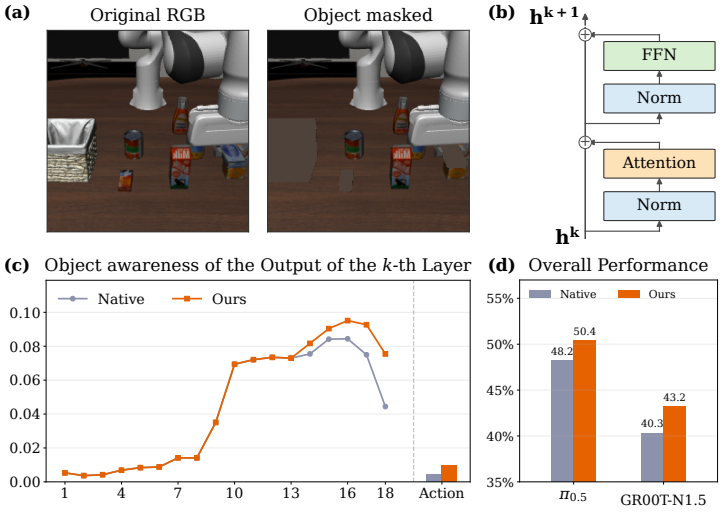

Figure 1: **A case study of layer-wise object awareness in VLA inference. (a)** Original and object-masked RGB observations in LIBERO-PRO, where task-relevant object regions are replaced with the mean RGB values of the image. **(b)** A Transformer layer in a VLA, mapping the action hidden state from h^k to h^{k+1} through a residual update. **(c)** Object awareness of $\pi_{0.5}$ layer outputs across depth under native inference and STAR. **(d)** Success rates of native inference and STAR on LIBERO-PRO for $\pi_{0.5}$ and GR00T-N1.5.

or fine-grained adjustment, the intervention should be conservative; these changes are reflected in recent proprioceptive trajectories. In short, STAR uses a proprioception-modulated EMA to smooth cross-layer residual updates, carrying object-aware directions across the mixing region.

This design is validated through our *object awareness* measurement and robotic manipulation benchmarks. In the object-awareness analysis, as shown in Figure 1 (c), STAR increases the object awareness of late-layer action representations. In robotic manipulation benchmarks, STAR enhances the performance of $\pi_{0.5}$ (Physical Intelligence et al., 2025) and GR00T-N1.5 (NVIDIA Research, 2025a) in LIBERO-PRO, as shown in Figure 1 (d), and improves success rates by an average of 4.0 percentage points across the four backbone–benchmark configurations in our main experiments.

Our contributions are summarized as follows:

- We introduce STAR, a training-free VLA inference method that steers action representations to improve visual grounding.
- Our technical innovation is an EMA that combines residual updates across late layers, together with a modulation strategy that adjusts its coefficient using proprioceptive states.
- Extensive experiments validate that STAR can consistently improve pre-trained VLA models across benchmarks, with an average success-rate improvement of 4.0 percentage points.

2 Related Work

2.1 Vision-Language-Action Models

Vision-Language-Action models (VLAs) are robot policies adapted from pretrained vision-language models (VLMs) with strong multimodal understanding abilities (Zitkovich et al., 2023; Zhang et al., 2026; Ma et al., 2026; Kawaharazuka et al., 2025; Sapkota et al., 2025). RT-2 (Zitkovich et al., 2023) established this paradigm by co-fine-tuning a VLM on vision-language data and robot trajectories, thereby aligning it with robotic control. This line was further developed by RoboFlamingo (Li et al., 2023a) and OpenVLA (Kim et al., 2024), which showed that VLM backbones can serve as strong and scalable policy priors. More recent works extend the action-token formulation (Zhong et al., 2025a) or the training pipeline while retaining this VLM-centric recipe. π_0 (Black et al., 2024) and $\pi_{0.5}$ (Physical Intelligence et al., 2025) adopt flow-based continuous action modeling. UniVLA (Bu et al., 2025) studies cross-embodiment policy learning through task-centric latent actions. The GR00T

series (NVIDIA et al., 2025; NVIDIA Research, 2025a;b; NVIDIA, 2026) balances efficiency and performance with a dual-system design. ReconVLA (Song et al., 2025) strengthens the perceptual role of VLA models by using reconstructive supervision. As VLA foundation models improve, our work explores a complementary training-free direction that further enhances their inference-time performance without additional data collection or parameter updates.

2.2 Training-free Methods in Embodied Policies

A growing body of work improves embodied policies at test time without updating model weights, with existing approaches differing mainly in where the intervention is applied. One line of work modifies **diffusion or flow trajectories**. TNB (Yoneda et al., 2023), HAJL (Luo et al., 2025), and DemoDiffusion (Park et al., 2025) re-noise and denoise alternative or retargeted action sequences, effectively blending them with the generative prior of a pre-trained policy. VLS (Liu et al., 2026) and DynaGuide (Du & Song, 2025) further steer denoising with guidance from additional vision foundation models. Another line targets **action decoding**. BID (Liu et al., 2024) restores reactivity via closed-loop resampling over candidate chunks. RTC (Black et al., 2025) enables asynchronous real-time control by freezing committed actions and inpainting the remaining chunk. Recent test-time scaling methods use online search or candidate selection to improve action choice, including VLA-Reasoner (Guo et al., 2025) and anti-exploration steering (Yang et al., 2025). A more recent line intervenes at the **spatial visual-token** level. IVRA (Park et al., 2026) refines relations among visual tokens using encoder-derived affinity hints, CGVD (Song et al., 2026) suppresses spatial distractors through concept-gated visual distillation, and DTP (Li et al., 2026) prunes task-irrelevant visual tokens that mislead action generation. Overall, these results highlight that performance gains can be achieved purely through inference-time modifications. Most approaches operate on explicit input-output interfaces. In contrast, our work explores whether directly intervening on transformer hidden-state dynamics can improve visual grounding and downstream control.

2.3 Visual Hallucination

Visual hallucination is common in multimodal models, including VLMs (Li et al., 2023b; Wang et al., 2023b; Guan et al., 2024; Wang et al., 2023a; Chen et al., 2026) and embodied visuomotor policies, where it manifests as weak visual grounding or action-level hallucination (Song et al., 2025; Yu et al., 2025; Soh & Lim, 2026). For VLMs, DAMO (Wang et al., 2025) mitigates visual hallucination by accumulating activation momentum during decoding. Similarly, DCLA (Tang et al., 2025) and VASparse (Zhuang et al., 2025) suggest that inference-time control of hidden states or visual token flow can improve faithfulness without retraining. For embodied policies, the analogous failure is inaccurate visual grounding, such as attending to distractors or acting on the wrong object. Knowledge Insulation (Driess et al., 2025) shows that naively adapting VLM backbones to action can erode the transferred semantic knowledge that supports grounding. ReconVLA (Song et al., 2025) strengthens grounding through a reconstructive objective over gaze regions, whereas DTP (Li et al., 2026) offers a lightweight training-free alternative by pruning distracting tokens at inference time. Our method is closest in spirit to DAMO, as both methods use training-free control of layer-wise dynamics. However, DAMO targets VQA tasks in VLMs, while our work targets robotic action execution. This embodied setting introduces proprioceptive execution states as a key signal for controlling the intervention, a feature specific to robot policies.

3 STAR: Steering Action Representations during Inference

3.1 Overview

Vision-language-action models generate actions from visual observations, proprioceptive states, and language instructions. At each control step t, a VLA policy receives an observation $o_t = (I_t, s_t)$, where I_t denotes the visual input and s_t denotes the robot proprioceptive state, together with a language instruction x. The policy outputs an action chunk:

$$A_t = [a_{t,1}, a_{t,2}, \ldots, a_{t,H_p}], \tag{1}$$

where H_p is the prediction horizon, and the first H_e actions $a_{t,1:H_e}$ are executed. Each executed action is decoded from the model's final action representations, making it important to preserve task-relevant visual evidence until the late layers.

Our key design principle is to preserve object-aware update directions across the late-layer mixing region using EMA. A fixed EMA strength, however, ignores the non-uniform dynamics of robotic manipulation. We characterize execution dynamics along a spectrum from *progress*, where the robot moves steadily along a consistent direction, to *transition*, where execution dynamics can change rapidly due to contact changes, end-effector reorientation, or fine-grained adjustment. Our heuristic is that transition phases require a more conservative EMA coefficient to limit the extent of inference-time editing, thereby preventing the policy from deviating too much from its original behavior. Following this principle, we propose STAR, a training-free VLA inference method that carries residual updates across the late-layer mixing region via a proprioception-modulated EMA. An overview of our method is illustrated in Figure 2.

3.2 EMA-Based Layer-wise Residual Update

We assume the VLA backbone contains n Transformer layers $L_1, \ldots, L_n$. We define the late-layer mixing region by a relative starting depth ρ:

$$p = \lfloor \rho n \rfloor, \qquad \mathcal{L}_{\text{early}} = \{L_1, \ldots, L_p\}, \qquad \mathcal{L}_{\text{mix}} = \{L_{p+1}, \ldots, L_n\}. \tag{2}$$

In our experiments, we set $\rho = 2/3$, so mixed updates are applied to the final third of the Transformer layers. This choice is motivated by the diagnostic study in Figure 1 and prior work on inference-time modification of multimodal models (Wang et al., 2025; Zhong et al., 2025b). The first $p = \lfloor \rho n \rfloor$ layers are left unchanged, allowing the backbone to form language-conditioned visual evidence. For layers in $\mathcal{L}_{\text{mix}}$, which are closer to action decoding and may become less object-aware, we apply proprioceptively modulated EMA updates.

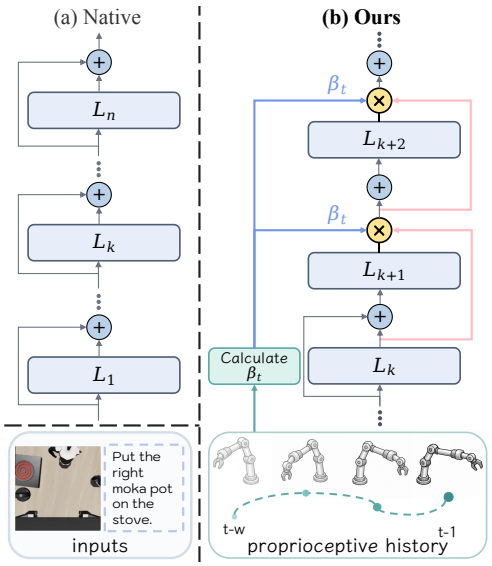

Figure 2: **Overview of action generation at step t.** STAR estimates a step-level EMA coefficient β_t from proprioceptive history and applies it to layer-wise updates. Blue arrows denote the EMA coefficient β_t derived from proprioceptive history, while pink arrows denote the cross-layer update momentum used in the EMA mixing.

Formally, we denote the original residual update of the k-th Transformer block at robot time step t as Δh_t^k, defined as its output minus its input before EMA mixing: $h_t^k = h_t^{k-1} + \Delta h_t^k$.

STAR mixes residual updates across late layers using an EMA. Analogous to momentum in optimization, it maintains a residual-update trend ("momentum") m_t^k across the late layers of each forward pass. At the first mixing layer, we initialize $m_t^{p+1} = \Delta h_t^{p+1}$ using that layer's original residual update. Subsequent layers ($k = p+2, \ldots, n$) update this trend using coefficient β_t:

$$m_t^k = \beta_t m_t^{k-1} + (1 - \beta_t)\Delta h_t^k. \tag{3}$$

The hidden states of layers in $\mathcal{L}_{\text{mix}}$ are updated using m_t^k rather than the original Δh_t^k:

$$\hat{h}_t^k = h_t^{k-1} + m_t^k, \qquad h_t^k \leftarrow \hat{h}_t^k, \quad L_k \in \mathcal{L}_{\text{mix}}. \tag{4}$$

This operation carries a smoothed update direction across depth, allowing object-aware information formed in earlier layers to persist through later layers.

Another question is how to choose β_t at each control step t. A fixed coefficient β_t cannot account for how robot execution alternates between *progress* and *transition* phases. We therefore compute β_t from causal proprioceptive evidence.

3.3 Causal Proprioceptive Evidence

A VLA policy operates in a closed sensorimotor loop, so the robot's own motion provides direct evidence of the current execution dynamics. Specifically, let $p_t = \text{pos}(s_t) \in \mathbb{R}^3$ denote the end-effector

position extracted from s_t, and let $R_t = \mathrm{rot}(s_t)$ denote the corresponding orientation representation. We construct four causal quantities from the end-effector trajectory and orientation history.

First, we measure directional continuity as $c_t = \cos(p_t - p_{t-1},\, p_{t-1} - p_{t-2})$. A high c_t indicates that the end-effector continues to move along a consistent direction, which often corresponds to steady progress.

Second, we measure the motion scale as $v_t = \|p_t - p_{t-1}\|_2$. This quantity reflects whether the robot is still undergoing substantial translational motion.

Third, we measure an acceleration-like change in translation as $j_t = \|(p_t - p_{t-1}) - (p_{t-1} - p_{t-2})\|_2$. This quantity captures abrupt changes in motion, which often occur near transitions, contact changes, or local corrections.

Fourth, we measure orientation change as $\omega_t = d_{\mathrm{rot}}(R_t, R_{t-1})$, where $d_{\mathrm{rot}}(\cdot,\cdot)$ denotes a rotation distance. Orientation changes are particularly informative for execution dynamics involving grasping, insertion, drawer interaction, or precise placement.

All quantities are computed only from observations available up to the current control step. Therefore, the modulation signal is strictly causal and does not use future states or oracle task information.

3.4 Causal Normalization

The raw magnitudes of proprioceptive quantities vary across embodiments, tasks, and simulators. We therefore normalize each scalar using a causal sliding window. For $x_t \in \{c_t, v_t, j_t, \omega_t\}$, we define the window length as $W = f_{\mathrm{ctrl}} T_w$ and the corresponding history as $\mathcal{H}_t^x = \{x_{t-W}, \ldots, x_{t-1}\}$, where f_{ctrl} is the control frequency and T_w is the temporal span of the statistics. The normalized value is then computed as

$$z_t^x = \frac{x_t - \mu(\mathcal{H}_t^x)}{\sigma(\mathcal{H}_t^x) + \epsilon}. \tag{5}$$

Since $\mathcal{H}_t^x$ only contains values before the current control step, the normalization remains strictly causal. This makes the normalized proprioceptive evidence depend on recent execution context rather than on the entire episode history.

3.5 Proprioceptive Mixing Score

We compute a signed proprioceptive mixing score from the normalized proprioceptive indicators:

$$r_t = w_c z_t^c + w_v z_t^v + w_j z_t^j + w_\omega z_t^\omega. \tag{6}$$

A larger normalized score leads to stronger cross-layer mixing, while a smaller one favors the current layer update. The sign of each coefficient determines whether the corresponding signal increases or decreases the late-layer cross-layer mixing strength. In our implementation, directional continuity and motion scale provide positive evidence for steady progress and thus stronger mixed updates, whereas abrupt translational change and orientation change provide negative evidence when they indicate transitions or fine adjustments.

To make the scale of r_t stable across tasks, we further apply a robust causal normalization. Let $\mathcal{R}_t = \{r_{t-W}, \ldots, r_{t-1}\}$ be the recent history of proprioceptive mixing scores. We compute:

$$\hat{r}_t = \frac{r_t - \mathrm{median}(\mathcal{R}_t)}{\mathrm{IQR}(\mathcal{R}_t) + \epsilon}. \tag{7}$$

The use of the median and interquartile range reduces sensitivity to transient spikes in proprioceptive measurements.

3.6 Proprioception-Conditioned Mixing Coefficient

The normalized proprioceptive score is mapped to the continuous coefficient used in the EMA mixed updates:

$$\beta_t = \beta_{\min} + (\beta_{\max} - \beta_{\min})\sigma(\kappa \hat{r}_t), \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function and κ controls the sharpness of the transition. A larger β_t preserves more accumulated update momentum in Eq. (3), helping late layers maintain object-aware update information during steady execution. A smaller β_t makes the cross-layer mixed update rely more on the current layer update, which is useful during transitions, contact changes, or fine adjustment.

This operation does not directly edit attention weights or prune visual tokens. Instead, it only changes how late-layer residual updates (He et al., 2016; Vaswani et al., 2017; Radford et al., 2019) are mixed before being propagated to the action head. Therefore, STAR directly targets the update stream where late-layer visual degradation occurs, while the proprioceptive coefficient β_t adapts the cross-layer mixing strength to the robot's current execution dynamics.

4 Experiments and Results

We evaluate STAR on two manipulation benchmarks, SimplerEnv (Li et al., 2024b) and LIBERO-PRO (Zhou et al., 2025). These benchmarks cover complementary aspects of VLA control: SimplerEnv emphasizes Google Robot-style tabletop manipulation with diverse object and articulated-object interactions, while LIBERO-PRO focuses on long-horizon desktop manipulation with compositional task variations. Our experiments are designed to answer three questions: (i) whether STAR improves frozen VLA policies without additional training, (ii) whether the proposed cross-layer residual mixing mechanism transfers across different VLA backbones, and (iii) which design choices are responsible for the improvement, including the proprioceptive evidence, beta scheduling, and intervention start layer.

4.1 Implementation Details

Backbones. We evaluate STAR on three representative VLA backbones with different vision-language architectures and action-generation heads. **GR00T-N1.5** (NVIDIA Research, 2025a) uses an NVIDIA Eagle VLM to encode language instructions and visual observations, and conditions a flow-based action head on the resulting vision-language embeddings together with robot state and action tokens. **OpenVLA** (Kim et al., 2024) is an open-source VLA model built on a Prismatic-style VLM. It combines DINOv2 and SigLIP visual encoders with a projection module and a Llama-2-based language backbone. We use OpenVLA-7B in our experiments. $\pi_{0.5}$ (Physical Intelligence et al., 2025) follows a PaliGemma-style VLM plus action-expert design. Its VLM backbone uses a SigLIP vision encoder and a Gemma-based language transformer, and the action expert generates continuous action chunks through flow matching. For each backbone, we compare the original frozen policy with the same policy equipped with STAR. All model weights remain frozen during evaluation.

Evaluation protocol. We apply STAR only at inference time and follow the official evaluation protocol of each benchmark. We report task success rate as the primary metric, including both per-task success rates and average success rates.

4.2 Benchmarks

SimplerEnv. SimplerEnv (Li et al., 2024b) evaluates policies in simulated Google Robot manipulation environments. We use six Fractal/Google Robot tasks. *Open Drawer* requires the robot to grasp and pull an articulated drawer handle. *Place in Drawer* requires opening a drawer, placing an object inside, and completing the placement under occlusion and articulated-object constraints. *Pick Coke Can* evaluates object-centric grasping and lifting for a commonly used can object. *Move Near* tests spatial relation reasoning by asking the robot to move one object close to another target object. *Close Drawer* requires pushing or sliding the drawer back to a closed state after interaction. *Pick Object* evaluates generic object picking across randomized object instances and placements. These tasks jointly cover grasping, articulated-object interaction, spatial relocation, and relation-aware control.

LIBERO-PRO. LIBERO-PRO (Zhou et al., 2025) evaluates long-horizon manipulation under multiple forms of generalization. We report five evaluation splits: *LIBERO-10-lan*, which paraphrases instructions while preserving task semantics; *LIBERO-10-object*, which modifies non-essential object attributes such as color, texture, or size; *LIBERO-10-swap*, which perturbs initial object placements and spatial configurations; *LIBERO-10-task*, which changes target objects, action requirements, or

Table 1: **Main results on SimplerEnv.** For OpenVLA, the average is computed over the available tasks. STAR improves frozen VLA policies on both backbones and achieves the best average success rate, outperforming both native inference and DTP without additional model forward passes.

Backbone	Method	SimplerEnv Google Robot Tasks						Avg.
		Open Drawer	Place in Drawer	Pick Coke Can	Move Near	Close Drawer	Pick Object	
GR00T-N1.5 (NVIDIA Research, 2025a)	Native	26.7%	4.0%	69.3%	68.7%	45.0%	32.3%	41.0%
	DTP	26.7%	4.0%	59.3%	68.7%	41.7%	39.3%	40.0%
	STAR (Ours)	**32.0%**	**4.7%**	**77.3%**	**76.7%**	**47.3%**	**40.0%**	**46.3%**
OpenVLA (Kim et al., 2024)	Native	7.4%	0.0%	65.3%	56.7%	22.2%	–	30.3%
	DTP	11.1%	0.0%	69.3%	**69.0%**	22.2%	–	34.3%
	STAR (Ours)	**18.5%**	**3.7%**	**70.0%**	61.7%	**25.9%**	–	**36.0%**

Table 2: **Main results on LIBERO-PRO.** STAR improves the average success rate on both GR00T-N1.5 and $\pi_{0.5}$, while DTP gives limited or mixed gains on this long-horizon benchmark.

Backbone	Method	LIBERO-PRO Perturbations on LIBERO-10					Avg.
		lan	object	swap	task	env	
GR00T-N1.5 (NVIDIA Research, 2025a)	Native	81.2%	40.4%	0.0%	6.0%	73.8%	40.3%
	DTP	82.0%	42.2%	0.0%	1.8%	75.6%	40.3%
	STAR (Ours)	**82.8%**	**48.0%**	**2.2%**	**6.8%**	**76.4%**	**43.2%**
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	Native	**92.8%**	66.2%	9.2%	6.8%	65.8%	48.2%
	DTP	90.4%	64.6%	8.0%	**7.8%**	67.6%	47.7%
	STAR (Ours)	**92.8%**	**68.6%**	**13.0%**	7.6%	**70.2%**	**50.4%**

task goals; and *LIBERO-10-env*, which varies visual context such as backgrounds, lighting, or textures. Each split contains ten tasks, and we report the success rate averaged over the tasks within each split.

4.3 Main Results

We compare STAR with native inference and a training-free attention-side baseline, DTP (Li et al., 2026). DTP aims to improve VLA inference by identifying task-relevant visual regions and pruning distracting image tokens according to action-token attention. It is therefore a natural baseline for testing whether improving visual attention allocation is sufficient for more reliable action generation. However, DTP requires an additional forward pass to obtain the attention signals used for pruning, whereas STAR requires no additional model forward passes beyond native inference. It directly modulates late-layer residual updates without editing attention weights, pruning visual tokens, or updating model weights.

Table 1 summarizes the SimplerEnv results. STAR improves the average success rate from 41.0% to 46.3% on GR00T-N1.5 and from 30.3% to 36.0% on OpenVLA. The gains are consistent across all six GR00T-N1.5 tasks and are stronger overall than those of DTP, showing that STAR is effective across the two backbone-specific evaluation settings. Table 2 reports the LIBERO-PRO results. STAR improves the average success rate from 40.3% to 43.2% on GR00T-N1.5 and from 48.2% to 50.4% on $\pi_{0.5}$. The largest gains occur on the object split for GR00T-N1.5 and the environment split for $\pi_{0.5}$. These results support the applicability of STAR across both autoregressive-style and continuous-action VLA backbones.

4.4 Ablation Studies

We conduct ablation studies with OpenVLA on SimplerEnv, and all ablations follow the same protocol as the main SimplerEnv evaluation. We focus on three factors: the proprioceptive evidence used to compute the proprioceptive mixing score, the beta scheduling strategy, and the layer region where modulation is applied. These ablations isolate the main design choices of STAR and test whether the observed gains come from meaningful proprioceptive modulation rather than from a generic late-layer perturbation.

Table 3: **Ablation on proprioceptive evidence.** Using all four proprioceptive cues gives the best average success rate, showing that mixed updates benefit from both steady-progress and transition-sensitive signals.

Variant	Proprioceptive Evidence				SimplerEnv Google Robot Tasks					Avg.
	c_t	v_t	j_t	ω_t	Open Drawer	Place in Drawer	Pick Coke Can	Move Near	Close Drawer	
Full STAR (Ours)	✓	✓	✓	✓	**18.5%**	**3.7%**	70.0%	**61.7%**	**25.9%**	**36.0%**
w/o direction consistency	×	✓	✓	✓	14.8%	0.0%	64.0%	56.7%	14.8%	30.1%
w/o translation scale	✓	×	✓	✓	11.1%	0.0%	66.7%	55.0%	22.2%	31.0%
w/o translational change	✓	✓	×	✓	14.8%	**3.7%**	65.3%	58.3%	14.8%	31.4%
w/o rotation change	✓	✓	✓	×	14.8%	0.0%	66.7%	58.3%	18.5%	31.7%
Positive evidence only	✓	✓	×	×	3.7%	0.0%	62.7%	**61.7%**	18.5%	29.3%
Transition evidence only	×	×	✓	✓	11.1%	**3.7%**	**70.7%**	58.3%	22.2%	33.2%
No proprioceptive evidence	×	×	×	×	11.1%	0.0%	56.0%	53.3%	18.5%	27.8%

Table 4: **Ablation on beta scheduling.** A proprioceptively scheduled β with a moderate range outperforms both fixed cross-layer mixing strengths and poorly calibrated dynamic ranges.

Variant	SimplerEnv Google Robot Tasks					Avg.
	Open Drawer	Place in Drawer	Pick Coke Can	Move Near	Close Drawer	
Fixed-$\beta = 0.25$	11.1%	0.0%	68.0%	56.7%	22.2%	31.6%
Fixed-$\beta = 0.35$	14.8%	0.0%	65.3%	56.7%	14.8%	30.3%
Fixed-$\beta = 0.45$	9.3%	1.9%	65.7%	58.3%	18.5%	30.7%
Fixed-$\beta = 0.55$	14.8%	0.0%	64.0%	60.0%	14.8%	30.7%
Dynamic-β $(0.15, 0.55)$	14.8%	0.0%	64.0%	55.0%	22.2%	31.2%
Dynamic-$\boldsymbol{\beta}$ $(\mathbf{0.25}, \mathbf{0.55})$ (Ours)	**18.5%**	**3.7%**	70.0%	61.7%	**25.9%**	**36.0%**
Dynamic-β $(0.35, 0.55)$	7.4%	0.0%	**70.7%**	51.7%	14.8%	28.9%
Dynamic-β $(0.25, 0.45)$	11.1%	0.0%	64.0%	53.3%	**25.9%**	30.9%
Dynamic-$\boldsymbol{\beta}$ $(\mathbf{0.25}, \mathbf{0.55})$ (Ours)	**18.5%**	**3.7%**	70.0%	61.7%	**25.9%**	**36.0%**
Dynamic-β $(0.25, 0.65)$	**18.5%**	**3.7%**	66.7%	**63.3%**	18.5%	34.1%

4.4.1 Effect of Proprioceptive Evidence

Table 3 studies the role of proprioceptive evidence. Removing any single cue causes a 4.3–5.9 percentage-point drop in average success rate, indicating that the four signals provide complementary information for characterizing execution dynamics across tasks. Direction consistency and translation scale help identify steady progress, while translational change and rotation change provide transition-sensitive evidence for contact changes and fine manipulation. The grouped ablations further support this interpretation: using only positive evidence reaches 29.3%, and using only transition evidence reaches 33.2%, both below the full model. The no-proprioception variant obtains 27.8%, confirming that the improvement is not simply caused by adding a late-layer mixing operation, but by conditioning the cross-layer mixing strength on meaningful robot-state dynamics.

4.4.2 Effect of Beta Scheduling

Table 4 compares fixed and dynamic choices of the cross-layer mixing coefficient. Fixed-β variants reach at most 31.6% average success, showing that a single cross-layer mixing strength cannot accommodate the different execution states in SimplerEnv. Dynamic scheduling improves performance when the range is properly calibrated: decreasing the lower or upper endpoint yields 31.2% with Dynamic-$\beta(0.15, 0.55)$ and 30.9% with Dynamic-$\beta(0.25, 0.45)$, respectively, while increasing them yields 28.9% with Dynamic-$\beta(0.35, 0.55)$ and 34.1% with Dynamic-$\beta(0.25, 0.65)$. The proposed

Table 5: **Ablation on start layer.** Applying mixed updates only to the final third of layers gives the best average success rate, supporting late residual updates as the effective intervention target.

Variant	SimplerEnv Google Robot Tasks					Avg.
	Open Drawer	Place in Drawer	Pick Coke Can	Move Near	Close Drawer	
$\rho = 0$	14.8%	0.0%	49.3%	53.3%	11.1%	25.7%
$\rho = 1/3$	11.1%	0.0%	55.0%	**68.3%**	3.7%	27.6%
$\rho = 2/3$ (Ours)	**18.5%**	**3.7%**	**70.0%**	61.7%	**25.9%**	**36.0%**
$\rho = 1$ (Native)	7.4%	0.0%	65.3%	56.7%	22.2%	30.3%

range, Dynamic-$\beta(0.25, 0.55)$, achieves the best average success rate of 36.0%, showing that both adaptivity and range calibration are necessary for effective proprioception-conditioned mixed updates.

4.4.3 Effect of Start Layer

Table 5 evaluates which layer region should use mixed updates. Mixing from the first layer reduces the average success rate to 25.7%, suggesting that early visual-language computation should remain unchanged. Starting after the first third of layers improves *Move Near* to 68.3%, but weakens object-centric and articulated-object manipulation, leading to a 27.6% average. Applying mixed updates only to the final third of layers achieves the best average success rate of 36.0% and the strongest performance on four out of five tasks. This supports the central design of STAR: intervene on the late residual update stream closest to action decoding, while preserving the backbone's native early-to-middle multimodal processing.

4.5 Real-Robot Experiments

We further evaluate STAR with $\pi_{0.5}$ on a real-robot task that requires picking up a pear and placing it in a bowl. We collect 100 episodes and fine-tune $\pi_{0.5}$ for 15,000 steps. Both native inference and STAR use the same fine-tuned checkpoint, with STAR applied only at inference time. Table 6 reports the success rates over 40 evaluation trials.

On the real-robot system with a single NVIDIA RTX 3090, STAR introduces only 3.3% additional inference latency relative to native inference.

Table 6: **Real-robot results.** Success rates on the pear-to-bowl task with $\pi_{0.5}$.

Method	Successes / Trials	Success Rate
Native	17/40	42.5%
STAR (Ours)	19/40	**47.5%**

5 Conclusion

We introduced STAR, a training-free method that uses a proprioception-modulated EMA to preserve object-aware residual updates in late VLA layers. It limits intervention during rapid execution changes. Experiments show improved late-layer object awareness and consistently higher average success rates across manipulation benchmarks. These results support residual steering as a complementary approach to improving visual grounding without retraining.

Limitations. While STAR improves training-free VLA inference across multiple benchmarks and backbones, several limitations remain. First, STAR requires access to intermediate Transformer hidden states. It therefore cannot be directly applied to black-box policies that expose only action outputs. Second, the proprioceptive cues used by STAR are designed primarily for single-end-effector manipulation tasks. Extending this design to more complex embodiments remains to be explored. Finally, STAR is based on an empirical observation that sensitivity to task-relevant objects declines in later VLA Transformer layers. We observe this pattern in the models evaluated here, but it is not theoretically guaranteed for all pretrained VLA models. STAR may offer limited benefits when this pattern is absent.

AI USE STATEMENT

Generative AI tools, specifically Large Language Models (LLMs), were used to improve the clarity and linguistic quality of the manuscript and to assist with code implementation. No LLMs were used for research idea generation or experimental design. All AI-assisted code was carefully reviewed and tested by the authors to ensure correctness, and all AI-assisted text was manually checked and revised. The authors take full responsibility for the final manuscript and experimental results.

ETHICS STATEMENT

Our evaluation is conducted on standard robotic manipulation benchmarks in simulation and on real-world robotic platforms under controlled experimental settings. The study does not involve human subjects, collection of personal or sensitive information, or the release of new datasets containing such information. Real-world experiments were carried out following standard laboratory safety and research integrity practices.

REPRODUCIBILITY STATEMENT

The backbone models and checkpoints used in this work are publicly available, and the experimental benchmarks are standardized and publicly accessible. We will release the implementation code in a public GitHub repository. Additional experimental results are provided in the appendix to further facilitate reproduction and verification of our findings.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Kevin Black, Noah Brown, James Darpinian, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Zhiyuan Chen, Yuecong Min, Jie Zhang, Bei Yan, Jiahao Wang, Xiaozhen Wang, and Shiguang Shan. A survey of multimodal hallucination evaluation and detection. *International Journal of Computer Vision*, 134(3):131, 2026.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 91–104, 2025.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.

Maximilian Du and Shuran Song. Dynaguide: Steering diffusion policies with active dynamic guidance. In *Advances in Neural Information Processing Systems*, 2025.

Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Multi-modal hallucination control by visual information grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14303–14312, June 2024.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14375–14385, 2024.

Wenkai Guo, Guanxing Lu, Haoyuan Deng, Zhenyu Wu, Yansong Tang, and Ziwei Wang. VLA-Reasoner: Empowering vision-language-action models with reasoning via online monte carlo tree search. *arXiv preprint arXiv:2509.22643*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. OPERA: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13418–13427, June 2024.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 2025.

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

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13872–13882, June 2024.

Chenyang Li, Jieyuan Liu, Bin Li, Bo Gao, Yilin Yuan, Yangfan He, Yuchen Li, and Jingqun Tang. DTP: A simple yet effective distracting token pruning framework for vision-language action models. *arXiv preprint arXiv:2601.16065*, 2026.

Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, Yan Peng, et al. Coa-vla: Improving vision-language-action models via visual-textual chain-of-affordance. *arXiv preprint arXiv:2412.20451*, 2024a.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action models via instruction-driven routing & sparsification. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, and Tao Kong. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023a.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, Singapore, 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20. URL https://aclanthology.org/2023.emnlp-main.20.

Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, et al. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint arXiv:2501.14818*, 2025b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Shuo Liu, Ishneet Sukhvinder Singh, Yiqing Xu, Jiafei Duan, and Ranjay Krishna. VLS: Steering pretrained robot policies via vision-language models. *arXiv preprint arXiv:2602.03973*, 2026.

Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Maximilian Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via closed-loop resampling. *arXiv preprint arXiv:2408.17355*, 2024.

Shengcheng Luo, Quanquan Peng, Jun Lv, Kaiwen Hong, Katherine Rose Driggs-Campbell, Cewu Lu, and Yong-Lu Li. Human-agent joint learning for efficient robot manipulation skill acquisition. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1370–1377. IEEE, 2025.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision–language–action models for embodied ai. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

NVIDIA. NVIDIA Isaac GR00T N1.7: Open reasoning vla model for humanoid robots. Hugging Face community article, April 2026. URL https://huggingface.co/blog/nvidia/gr00t-n1-7. Accessed: 2026-05-05.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

NVIDIA Research. GR00T N1.5: An improved open foundation model for generalist humanoid robots. Technical blog / model release, June 2025a. URL https://research.nvidia.com/labs/gear/gr00t-n1_5/. Accessed: 2026-05-05.

NVIDIA Research. GR00T N1.6: An improved open foundation model for generalist humanoid robots. Technical blog / model release, December 2025b. URL https://research.nvidia.com/labs/gear/gr00t-n1_6/. Accessed: 2026-05-05.

Anupam Pani and Yanchao Yang. Gaze-regularized vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2603.23202*, 2026. URL https://arxiv.org/abs/2603.23202.

Jongwoo Park, Kanchana Ranasinghe, Jinhyeok Jang, Cristina Mata, Yoo Sung Jang, and Michael S Ryoo. IVRA: Improving visual-token relations for robot action policy with training-free hint-based guidance. *arXiv preprint arXiv:2601.16207*, 2026.

Sungjae Park, Homanga Bharadhwaj, and Shubham Tulsiani. Demodiffusion: One-shot human imitation using pre-trained diffusion policy. *arXiv preprint arXiv:2506.20668*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

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

Sangmim Song, Sarath Kodagoda, Marc Carmichael, and Karthick Thiyagarajan. Overcoming visual clutter in vision language action models via concept-gated visual distillation. *arXiv preprint arXiv:2603.10340*, 2026.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. ReconVLA: Reconstructive vision-language-action model as effective robot perceiver. *arXiv preprint arXiv:2508.10333*, 2025.

Kai Tang, Jinhao You, Xiuqi Ge, Hanze Li, Yichen Guo, and Xiande Huang. Mitigating hallucinations via inter-layer consistency aggregation in large vision-language models. *arXiv preprint arXiv:2505.12343*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Khoa Vo, Taisei Hanyu, Yuki Ikebe, Trong Thang Pham, Nhat Chung, Minh Nhat Vu, Duy Nguyen Ho Minh, Anh Nguyen, Anthony Gunderman, Chase Rainwater, et al. Clutter-resistant vision-language-action models through object-centric and geometry grounding. *arXiv preprint arXiv:2512.22519*, 2025.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, et al. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023a.

Junyang Wang, Yiyang Zhou, Guohai Xu, Pengcheng Shi, Chenlin Zhao, Haiyang Xu, Qinghao Ye, Ming Yan, Ji Zhang, Jihua Zhu, et al. Evaluation and analysis of hallucination in large vision-language models. *arXiv preprint arXiv:2308.15126*, 2023b.

Kaishen Wang, Hengrui Gu, Meijun Gao, and Kaixiong Zhou. DAMO: Decoding by accumulating activations momentum for mitigating hallucinations in vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Siyuan Yang, Yang Zhang, Haoran He, Ling Pan, Xiu Li, Chenjia Bai, and Xuelong Li. Steering vision-language-action models as anti-exploration: A test-time scaling approach. *arXiv preprint arXiv:2512.02834*, 2025.

Takuma Yoneda, Luzhe Sun, Ge Yang, Bradly C. Stadie, and Matthew R. Walter. To the noise and back: Diffusion for shared autonomy. In *Robotics: Science and Systems*, 2023.

Hang Yu, Juntu Zhao, Yufeng Liu, Kaiyu Li, Cheng Ma, Di Zhang, Yingdong Hu, Guang Chen, Junyuan Xie, Junliang Guo, Junqiao Zhao, and Yang Gao. Point what you mean: Visually grounded instruction policy. *arXiv preprint arXiv:2512.18933*, 2025.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. Vlm4vla: Revisiting vision-language-models in vision-language-action models. *arXiv preprint arXiv:2601.03309*, 2026.

Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025a.

Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. Aim: Adaptive inference of multi-modal llms via token merging and pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20180–20192, 2025b.

Jiaying Zhou, Zhihao Zhan, Ruifeng Zhai, Qinhan Lyu, Hao Liu, Keze Wang, Liang Lin, and Guangrun Wang. TAG: Target-agnostic guidance for stable object-centric inference in vision-language-action models. *arXiv preprint arXiv:2603.24584*, 2026.

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.

Xianwei Zhuang, Zhihong Zhu, Yuxin Xie, Liming Liang, and Yuexian Zou. VASparse: Towards efficient visual hallucination mitigation for large vision-language model via visual-aware sparsification. *arXiv preprint arXiv:2501.06553*, 2025.

Brianna Zitkovich, Nikhil Joshi, Aviral Lohan, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, 2023.

A INTERNAL MECHANISM OF STAR

We have shown in Section 4 that STAR improves VLA performance at inference time. This section reveals the internal effect of STAR.

A.1 EXPERIMENT SETUP

All experiments in this section are conducted with the $\pi_{0.5}$ VLA model on the LIBERO-PRO benchmark. We record two quantities at layer k. First, we record the hidden state of the action expert after the layer, denoted by h^k. Second, we record the action-to-image attention map, denoted by A^k. For masking diagnostics, we compare the original unmasked visual input with masked visual inputs obtained by replacing pixels inside a selected mask region with the mean RGB value of the image, while keeping all other inputs unchanged. This masking operation removes local visual evidence. Figure 3 summarizes the complete diagnostic protocol and the main empirical observations used in this section.

A.2 OBJECT AWARENESS OF ACTION HIDDEN STATES

A VLA policy ultimately predicts actions from action representations rather than from attention visualizations. Therefore, the diagnostic question we ask is not only whether the model visually attends to the target object, but whether the task-relevant object evidence actually affects the hidden state used for action generation. To measure this effect, we define an awareness score based on object masking.

For a mask region m, let $h^k_{\varnothing}$ be the layer-k hidden state under the original visual input, and let h^k_m be the hidden state under the corresponding masked input. We define the layer-wise awareness induced by mask m as

$$\mathcal{A}^k(m) = \frac{\left\| h^k_{\varnothing} - h^k_m \right\|_2}{\left\| h^k_{\varnothing} \right\|_2}. \tag{9}$$

When m is the task-relevant object mask m_o, the resulting quantity $\mathcal{A}^k(m_o)$ is the object awareness of the layer. A larger value means that masking the task object produces a larger relative change in the action hidden state, indicating stronger sensitivity to task-relevant object evidence.

In Section 1, we introduce object awareness in its simplest form, using the task-object mask: $\|h^k_{(\mathrm{m})} - h^k_{(\mathrm{o})}\| / \|h^k_{(\mathrm{o})}\|$. We now generalize the same idea into a mask-dependent awareness function to conduct controlled experiments. Under this definition, $\mathcal{A}^k(m_o)$ recovers the object-awareness score introduced in Section 1.

To verify that object awareness is not merely measuring generic sensitivity to image corruption, we construct two controls: the same number of pixels randomly scattered across the image (m_r), and a rectangular region of similar area randomly sampled from the background (m_b). We define control awareness as

$$\mathcal{A}^k_{\mathrm{ctrl}} = \max\left\{ \mathcal{A}^k(m_b), \mathcal{A}^k(m_r) \right\}. \tag{10}$$

To summarize the masking comparison, we use two statistics: object-awareness gain and object-selectivity ratio. They quantify two complementary effects: whether STAR increases sensitivity to the task-relevant object compared with native inference, and whether the measured sensitivity is stronger for the task-relevant object than for same-area control regions.

The object-awareness gain compares STAR with native inference:

$$R^k_{\mathrm{gain}} = \frac{\mathcal{A}^k_{\mathrm{STAR}}(m_o)}{\mathcal{A}^k_{\mathrm{native}}(m_o)}. \tag{11}$$

The object-selectivity ratio compares object masking with same-area control masking under the same inference method $u \in \{\mathrm{native}, \mathrm{STAR}\}$:

$$R^k_{\mathrm{select}}(u) = \frac{\mathcal{A}^k_u(m_o)}{\mathcal{A}^k_{u,\mathrm{ctrl}}}. \tag{12}$$

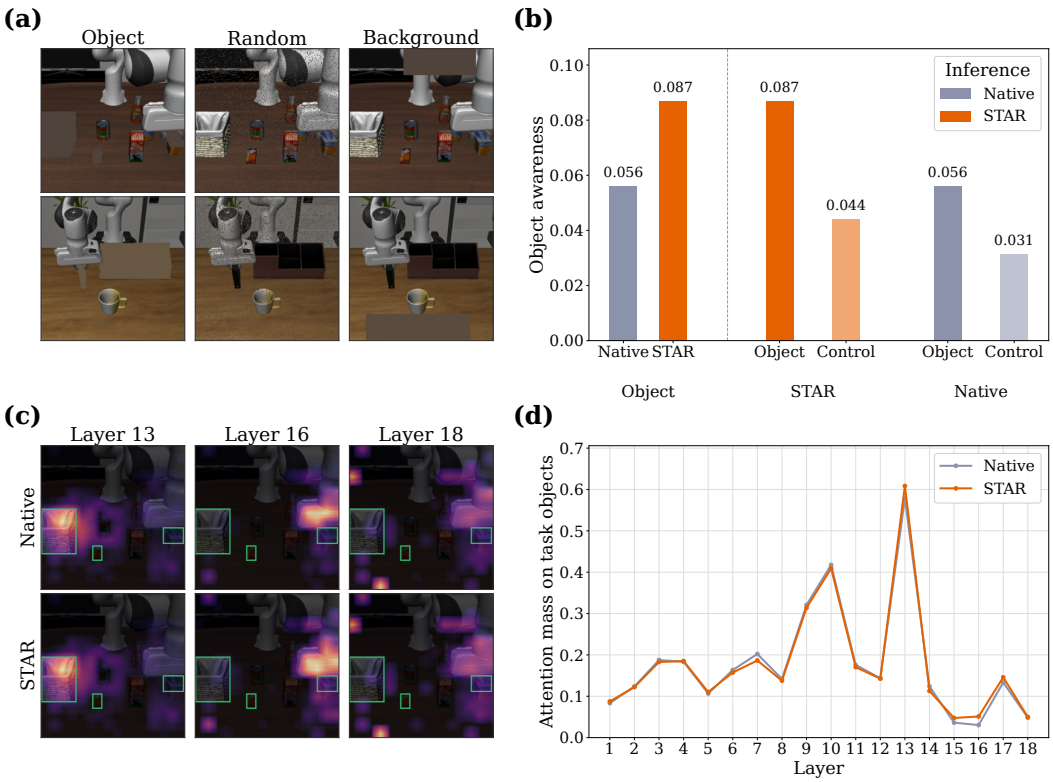

Figure 3: Object-awareness and attention-map diagnostics for native inference and STAR. (a) Visualization of the three masks, including the task-specific object mask m_o, randomly scattered pixels m_r with the same pixel count, and a randomly sampled rectangular background region m_b of similar area. (b) Comparison of awareness scores for native inference and STAR under object masking and control masking at layer 18. (c) Attention-map visualization at layers 13, 16, and 18, showing that native inference and STAR produce similar attention patterns. (d) Attention mass on the task-object region across layers. Native inference and STAR exhibit close attention-mass curves, indicating that attention-based object focus does not clearly reveal the hidden-state intervention effect.

Thus, $R^k_{\text{gain}} > 1$ indicates that STAR increases object awareness relative to native inference, while $R^k_{\text{select}} > 1$ indicates that the measured awareness is stronger for the task object than for same-area control regions.

Using the above metric, we compare native inference and STAR on the paired original and masked inputs. Figure 3(a) visualizes the three mask types used in the diagnostic, and Figure 3(b) reports the corresponding average awareness scores at layer 18 under native inference and STAR. Under the experimental setting with the $\pi_{0.5}$ VLA model on the LIBERO-PRO benchmark, STAR increases object awareness from 0.056 to 0.087, giving $R_{\text{gain}} = 1.554$.

The selectivity statistics further show that the measured awareness is task-object specific. For native inference, $R_{\text{select}}(\text{native}) = 1.806$. For STAR, $R_{\text{select}}(\text{STAR}) = 1.977$. Therefore, the increased awareness under STAR is not merely a response to arbitrary masking; it is stronger when the task-relevant object region is removed than when same-area control regions are removed.

These results show that object awareness is an effective diagnostic for the hidden-state intervention. It clearly captures that STAR makes late action representations more sensitive to task-relevant object evidence. As shown later in Section A.4, this object-awareness signal is much clearer than the corresponding attention-map change.

A.3 Gradient-based Saliency Map

The masking analysis above measures whether the action hidden state is sensitive to the task object. We further ask a complementary, action-grounded question: does the final predicted action rely more strongly on pixels belonging to that object? To answer this question, we use input-gradient saliency (Simonyan et al., 2013), which identifies the image evidence that locally influences the action output.

For a predicted action chunk $a_{1:K}$, we first average the actions in the chunk,

$$u = \frac{1}{K} \sum_{k=1}^{K} a_k, \tag{13}$$

and construct a scalar action score by projecting u onto its stop-gradient direction:

$$s = u^\top \mathrm{sg}\left(\frac{u}{\|u\|_2 + \epsilon} \right), \tag{14}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. Holding the action direction fixed during differentiation gives a scalar whose input gradient measures how each image region supports the action selected by the policy, without allowing the reference direction itself to change during backpropagation.

Let I be the input image and let $g(p)$ denote the magnitude of $\partial s / \partial I$ at pixel p, aggregated over the color channels. We spatially normalize this evidence and define the target-object gradient score as

$$\mathcal{S}_{\mathrm{grad}} = \frac{\sum_{p \in \Omega_o} g(p)}{\sum_{p \in \Omega_{\mathrm{img}}} g(p) + \epsilon}, \tag{15}$$

where Ω_o is the task-object region and Ω_{img} contains all image pixels. Thus, a larger $\mathcal{S}_{\mathrm{grad}}$ means that a greater fraction of the action-relevant visual evidence lies on the key object.

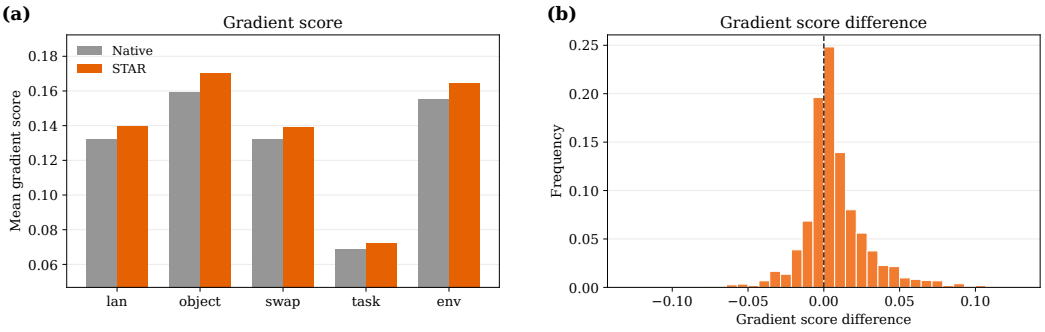

Figure 4: **Action-gradient scores on LIBERO-PRO.** (a) Mean normalized input-gradient evidence contained in the task-object region for native inference and STAR across the five LIBERO-PRO evaluation splits. Higher values indicate that the predicted action depends more strongly on the key object. STAR consistently increases the score on all five splits. (b) Distribution of paired score differences, computed as STAR minus native inference. Most of the mass lies to the right of zero, showing that the increase is consistent across evaluation samples rather than being driven only by split-level averages.

Figure 4(a) shows that STAR achieves a higher target-object gradient score than native inference on every LIBERO-PRO split. The improvement is therefore not restricted to a particular perturbation type: it appears under language, object-attribute, spatial-configuration, task, and visual-context variation. The paired differences in Figure 4(b) are also predominantly positive, indicating that the aggregate gains reflect a broadly consistent sample-level trend. These results are consistent with the final action depending more strongly on task-object pixels under STAR.

The qualitative maps in Figure 5 reveal the corresponding spatial effect. Under native inference, the action gradients are often weaker on the task object or dispersed over the robot and background. With STAR, stronger evidence is concentrated inside the cyan task-object regions, including articulated

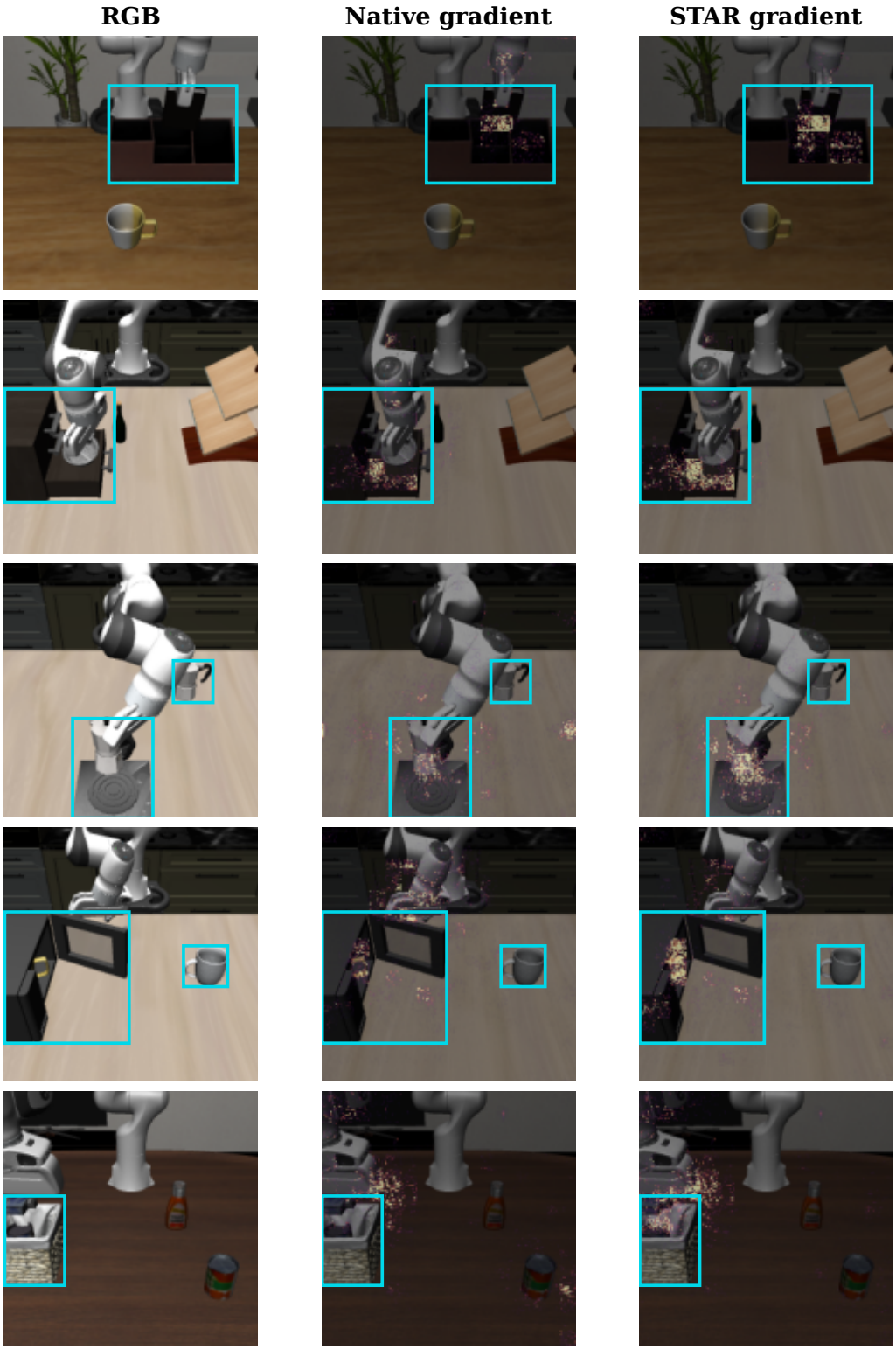

Figure 5: **Qualitative action-gradient saliency maps.** The left column shows the RGB observations with task-relevant objects marked by cyan boxes; the middle and right columns show saliency under native inference and STAR, respectively. Warmer pixels exert a stronger local influence on the predicted action. Compared with native inference, STAR concentrates more action-relevant evidence within the marked object regions across diverse manipulation scenes.

fixtures, manipulated objects, and interaction targets. Importantly, this result complements the attention analysis in Section A.4: STAR need not visibly reallocate attention mass to make the action more object-grounded. Instead, by carrying object-aware residual-update directions into the late action representations, it increases the local sensitivity of the final action to key-object pixels. Together, the masking and gradient diagnostics support the proposed mechanism: STAR preserves task-relevant visual evidence in the hidden-state update stream and makes that evidence more consequential for action generation.

A.4 Attention Maps Do Not Reveal the Intervention Effect

We next test whether attention maps can reveal the effect of the hidden-state intervention. We compare native inference and STAR using the same input and record the action-to-image attention maps at late action-expert layers. Figure 3(c) visualizes representative attention maps from native inference and STAR at layers 13, 16, and 18. Although STAR modifies the hidden-state stream, the corresponding attention maps remain visually similar across these layers.

To provide a more quantitative comparison, we compute the attention mass assigned to the task-object region. Let Ω_o denote the image-token set covered by the task-object mask, Ω_{img} denote all image tokens, and A^k denote the attention map of layer k. For method $u \in \{\mathrm{native}, \mathrm{STAR}\}$, the layer-wise object attention mass is

$$\mathcal{M}_u^k = \frac{\sum_{p\in\Omega_o} A_u^k(p)}{\sum_{p\in\Omega_{\mathrm{img}}} A_u^k(p)}. \tag{16}$$

Figure 3(d) reports $\mathcal{M}_u^k$ across layers for native inference and STAR. These results show that the attention-based measure of object focus remains close to that of native inference and does not reliably expose the hidden-state intervention.

A.5 Diagnostics of STAR's Internal Mechanism

The diagnostics above suggest that the internal effect of STAR is primarily reflected in the action hidden state and its dependence on visual evidence rather than in a visible reallocation of action-to-image attention. Object masking shows that STAR increases the sensitivity of late action representations to task-relevant object evidence, while same-area control masks induce smaller responses. Gradient saliency further shows that a larger fraction of the visual evidence influencing the final action is concentrated on the task object. In contrast, the attention maps and attention-mass curves remain close to native inference. Thus, STAR preserves task-object information in late action representations and makes it more consequential for action generation without necessarily producing a salient change in attention visualizations.