A^3-VLA: Automatic Perception-Guided Attention Alignment for Robot Manipulation

Shengzhe Zhang[1] **Qi Zhang**[2*] **Dazhong Shen**[3]
Chao Wang [4,5*] **Shaopeng Zhai** [2] **Hui Xiong** [6,7*]

[1]School of Computer Science and Technology , University of Science and Technology of China
[2]Shanghai AI Laboratory
[3]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[4]School of Artificial Intelligence and Data Science, University of Science and Technology of China
[5]State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
[6]Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou)
[7]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology

owen_zsz@mail.ustc.edu.cn, zhangqi.fqz@gmail.com, dazh.shen@gmail.com,
wangchaoai@ustc.edu.cn, zsp1197@163.com, xionghui@ust.hk

Abstract

Vision-Language-Action Models (VLAs) enable robots to perform manipulation tasks by leveraging Vision-Language Models (VLMs) to process visual observations and language instructions. However, the action prediction objective itself lacks fine-grained task-specific perceptual supervision, making it prone to relying on task-irrelevant visual shortcuts, leading attention away from target objects and key interaction areas, thereby limiting its operational performance and generalization ability. To address this, we propose Automatic Perception-Guided Attention Alignment for Robot Manipulation (A^3-VLA), a framework that derives task-specific visual guidance from the manipulation process itself. Specifically, A^3-VLA first generates heuristic candidate regions for target objects and the robot from interaction-induced visual changes, and refines them with a visual segmentation model offline. These regions serve as explicit perceptual guidance through two complementary objectives: an attention allocation loss that encourages sufficient focus on task-critical regions when language or robot state tokens query the visual input, and a contrastive learning loss that aligns the representations of candidate target regions with language instructions while separating them from task-irrelevant regions. Extensive experiments on simulated and real-world manipulation tasks demonstrate that A^3-VLA consistently improves multiple VLA backbones, with an average success rate gain of 6.5% - 25.2% across benchmarks.

1 Introduction

Embodied manipulation aims to enable robots to interact with the environment through learned policies to complete manipulation tasks [9, 1]. A core challenge is enabling robots to understand both what to manipulate (from language instructions) and how to manipulate (from visual observations). Recently, Vision-Language Models (VLMs) have achieved remarkable success in aligning visual and linguistic representations, demonstrating strong capabilities in tasks such as visual question answering and cross-modal reasoning [31, 2, 34, 33]. Building upon these advances, Vision-Language-Action

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Models (VLAs) [45, 27, 4] extend VLMs to robot control by directly generating action sequences conditioned on visual observations and language instructions [22, 6, 24, 36].

Early works on VLAs primarily focus on accurate action generation, pioneered by tokenization-based [27, 16] and flow-matching-based [4, 20] paradigms. Despite this progress in action learning, the visual perception capability of these models, including their ability to attend to regions specified by language instructions, remains fragile. As shown in Figure 1, these cases reveal a key failure mode that is often overlooked in VLA research: the model fails because the VLM backbone's attention mechanism fails to focus on the task-relevant regions specified by the language instruction, rather than being solely attributable to action generation errors. Thus, the VLA learns to "memorize" specific action trajectories from the training data, rather than connecting the instructions with the observation of the target objects. To break this pattern, VLAs need explicit signals about where to look, raising a natural question: can we obtain such task-specific guidance from the manipulation process itself?

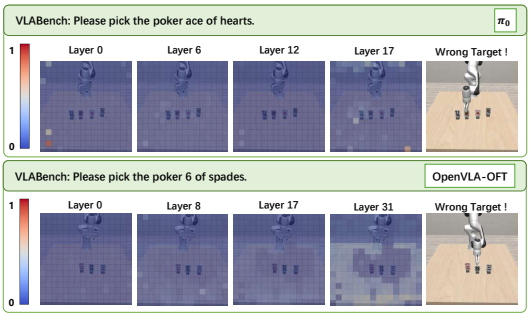


Figure 1: A failure case of the attention maps of language instruction querying visual observations in π_0 and OpenVLA-OFT on a Select Poker task from VLABench, visualized across layers from shallow (left) to deep (right).

We argue that the above limitation reveals two deeper issues with how VLAs currently handle visual perception. First, the fine-tuning process provides no explicit signal indicating which image regions are critical for the given task. The flow-matching or autoregressive objective only supervises action accuracy [40, 11, 41]; whether the model's visual attention aligns with task-relevant regions is never directly evaluated or encouraged. As a result, the VLM backbone's pretrained ability to discriminatively attend to semantically meaningful regions can gradually drift during fine-tuning [28, 18, 35]. Second, prior works attempt to enhance perception in different ways, such as CoT-VLA [43] with visual chain-of-thought reasoning, $\pi_{0.5}$ [12] and Gemini Robotics [26] with open-world knowledge. The guidance they provide remains either indirect (reasoning over the entire scene) or task-agnostic (general knowledge from heterogeneous data). These motivate a different perspective: the manipulation process itself encodes rich information about what matters visually. The moment of physical interaction, the trajectory of the robot end-effector, and the target object being acted upon all carry signals about which image regions deserve focused attention. If we obtain these signals and utilize them as explicit perceptual guidance, the task-oriented visual perception capability may be preserved during fine-tuning, rather than being overwritten by the action learning objective.

To address these challenges, we propose **A**utomatic Perception-Guided **A**ttention **A**lignment for Robot Manipulation (A^3-VLA), a framework that provides explicit visual perception guidance during VLA fine-tuning. The key insight of A^3-VLA is that interaction moments in the manipulation process naturally reveal the locations of task-critical regions. Based on this, A^3-VLA consists of two stages. First, a heuristic region proposal and temporal refinement stage extracts noisy candidate regions from interaction-induced visual changes and refines them with SAM2 into temporally consistent pseudo regions for target objects and the robot. Second, a perception-guided fine-tuning stage incorporates these pseudo regions via two complementary objectives: (i) an attention allocation loss that encourages the model to focus sufficient attention on the heuristic regions of target objects or the robot when language or robot state tokens query the visual input respectively, and (ii) a contrastive learning loss that aligns the representations of target object regions with language instructions while pushing away those of task-irrelevant regions. Together, these objectives help the VLA maintain task-focused visual attention throughout fine-tuning, rather than allowing it to drift toward task-irrelevant regions. Overall, our contributions are as follows:

- We propose A^3-VLA, a framework that derives task-specific visual guidance from the manipulation process itself, eliminating manual annotation.
- We design two auxiliary training objectives, attention allocation and contrastive representation alignment, that explicitly steer VLA visual focus toward task-relevant regions.

- Extensive experiments on both simulation and real-world settings demonstrate that A^3-VLA consistently improves multiple VLA backbones across diverse manipulation tasks.

2 Related Work

In this section, we provide previous research relevant to our work, including Vision-Language-Action Models (VLAs) and multimodal perception.

2.1 Vision-Language-Action Models (VLAs)

Foundational works such as RT-1 [5] and RT-2 [45] pioneered action tokenization for end-to-end policy learning, followed by open-source VLAs including Octo [27] and OpenVLA [16], and diffusion/flow-matching approaches like Gr00t series [3] and π_0 [4]. Recent extensions have explored efficient fine-tuning recipes [10, 23], and lightweight adaptation modules [29] to enhance the efficiency of VLAs. Across these methods, primary VLAs mainly supervise action learning yet overlook the significance of multimodal perception during fine-tuning [24], which makes language following and task understanding remain challenging for VLAs [13, 37].

2.2 Multimodal Perception for VLA

VLMs have demonstrated strong multimodal perception capabilities in tasks including VQA [8, 44], reasoning [43, 38], and cross-modal generation [30, 39], etc. Some works on VLAs attempt to explore and utilize the multimodal perception capability of VLMs to enrich the scene and task understanding of embodied models. For instance, RoboCLIP [25] and Embodied CLIP [14] ground language in robotic affordances with contrastive learning on the context of egocentric videos. FPC-VLA [32] utilizes a fine-tuned supervisor to check for failure based on multimodal observations. Recently, $\pi_{0.5}$ [12] pretrains VLAs with heterogeneous data sources to transfer open-world multimodal knowledge. Gemini Robotics [26] harnesses Gemini's capacity for physical reasoning to advance spatial comprehension. Unlike prior methods that rely on general multimodal knowledge or indirect scene-level supervision, A^3-VLA automatically derives task-specific interaction cues from manipulation trajectories and refines them with a visual segmentation model into soft pseudo regions for region-level perception guidance during fine-tuning.

3 Methodology

In this section, we introduce the framework of Automatic Perception-Guided Attention Alignment for Robot Manipulation (A^3-VLA), which automatically derives pseudo-region guidance from the manipulation process and delivers explicit guidance for multimodal perception. Its framework is illustrated in Figure 2. The key components of A3-VLA are: 1) **heuristic region proposal**, which extracts noisy candidate regions for target objects and the robot from demonstration trajectories and refines them with SAM2 [21]; 2) **attention weight allocation**, which encourages sufficient attention mass on pseudo regions; 3) **contrastive learning**, which aligns candidate target-region representations with language instructions; and 4) **joint VLA training** with the main action objective and auxiliary perceptual objectives.

3.1 Preliminary

The VLA flow model [4] is a powerful and widely accepted paradigm for action generation, adopted as the backbone of our A^3-VLA. Formally, the observation at the timestep t is denoted as $o_t = [I_1^t, ..., I_n^t, l^t, s^t]$, where I_i^t is the i-th view of images, l^t is the language instruction, and s^t is the current robot state. The main goal for flow matching-based VLAs is to learn the future action chunk $A_t = \{a_t, ..., a_{t+H}\}$, where a_t and H represent the action and the action horizon respectively. The corresponding flow matching loss L_{FM} is formulated as follows:

$$L_{FM}(\theta) = \mathbb{E}_{p(\mathbf{A}_t|\mathbf{o}_t), q(\mathbf{A}_z^t|\mathbf{A}_t)} \left[\left\| \mathbf{f}_\theta(\mathbf{A}_z^t, \mathbf{o}_t) - \mathbf{u}(\mathbf{A}_z^t|\mathbf{A}_t) \right\|^2 \right], \tag{1}$$

where θ is the parameter of the model, $\mathbf{A}_z^t$ is the noised action whose noise distribution is $q(\mathbf{A}_z^t|\mathbf{A}_t) = \mathcal{N}(zA_t, (1-z)I)$, $\mathbf{f}_\theta(\mathbf{A}_z^t, \mathbf{o}_t)$ denotes the output of the VLA, and $\mathbf{u}(\mathbf{A}_z^t|\mathbf{A}_t)$ is the denoising vector field in flow matching.

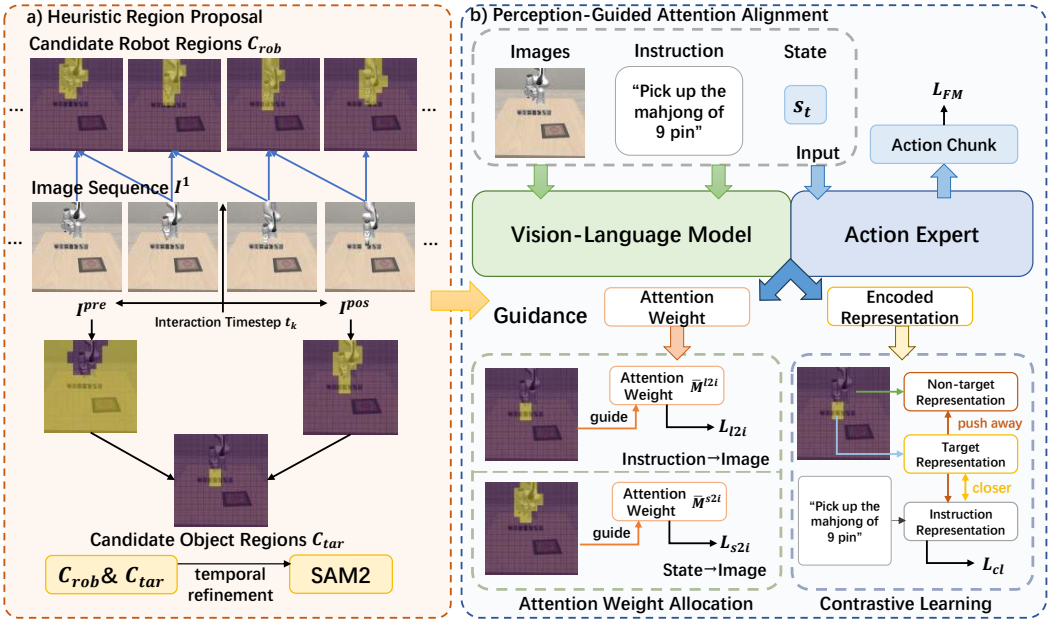


Figure 2: Overview of the A^3-VLA framework.

3.2 Heuristic Region Proposal

To provide explicit perceptual guidance during VLA fine-tuning, we first propose candidate regions that are likely to be task-relevant for the given manipulation task. Our key insight is that useful heuristic cues can be naturally extracted from the manipulation process itself. In many manipulation tasks, the robot physically interacts with task-relevant objects during the trajectory. These interaction moments, which are detectable from the robot states or sensors, provide direct signals for localizing both the target objects and the robot in the visual observation. In the following, we describe how interaction cues are used to generate candidate target and robot regions.

Formally, consider a trajectory with a sequence of images $I = \{I^1, ..., I^T\}$ and corresponding robot state sequence $S = \{s^1, ..., s^T\}$, where T is the trajectory length. The interaction timestep t_k denotes a coarse moment when the robot starts interacting with the task-relevant object. In practice, such moments can be obtained from robot or task cues, e.g., gripper states, contact signals, or predefined state ranges. Once t_k is determined, we partition image patches into two key region sets: the target object C_{tar} and the robot C_{rob}. The two region types are identified through different procedures, reflecting their distinct roles in the manipulation process. For target object regions C_{tar}, to localize the target object, we leverage the fact that it remains in place before the robot makes contact but undergoes visual changes once interaction begins. We define a window of w frames, and for each patch at spatial position (i, j), we compute the mean absolute difference between consecutive frames within the pre-interaction window $[t_k - 2w, t_k - w]$ and the post-interaction window $[t_k, t_k + w]$ respectively:

$$d^{pre}_{(i,j)} = \frac{1}{w} \sum_{t=t_k-2w}^{t_k-w} ||I^{t+1}_{(i,j)} - I^t_{(i,j)}||_2, \tag{2}$$

$$d^{post}_{(i,j)} = \frac{1}{w} \sum_{t=t_k}^{t_k+w} ||I^{t+1}_{(i,j)} - I^t_{(i,j)}||_2, \tag{3}$$

where $I^t_{(i,j)}$ denotes the RGB value of patch (i, j) at the t-th frame. This heuristic relies on the observation that the target object is often visually stable before physical interaction and changes after being manipulated. Then, patches with low pre-interaction variation but high post-interaction variation, i.e., $d^{pre}_{(i,j)} < \tau$ and $d^{post}_{(i,j)} > \tau$, are treated as candidate regions for the target object C_{tar}. These patches are visually stable before interaction but change once the robot begins interacting, matching the expected behavior of an object being manipulated. For the robot regions C_{rob}, unlike

target objects, it is present and active throughout the entire trajectory. It is therefore identified by finding patches with substantial variation across nearby observations:

$$d^t_{(i,j)} = ||I^{t+w}_{(i,j)} - I^t_{(i,j)}||_2. \tag{4}$$

Then, patches with high variation, i.e., $d^t_{(i,j)} > \tau$, are treated as candidate regions C_{rob}. Here τ is a variation threshold. We use $\tau = 10$ across all experiments and find performance is insensitive to this choice. The above description assumes a single interaction per trajectory for clarity. For tasks involving multiple interactions, the trajectory is partitioned at each interaction timestep, regions are obtained for each segment using the same method, and the final label sets are taken as the union across all segments.

Notably, a key practical advantage of this heuristic region proposal approach is that it requires no manual labeling, since it operates solely on the demonstration trajectory. These heuristic proposals are not intended to be accurate object masks. Instead, they serve as high-recall pseudo regions that provide coarse task-relevant visual cues. The asymmetric criterion, i.e., low variation before interaction and high variation after interaction, helps suppress patches that are consistently dynamic or unrelated to the contact event. For multi-interaction tasks, we apply the same procedure within each interaction segment before taking the union of the resulting regions, which reduces the ambiguity caused by temporally separated object changes.

Since the heuristic proposals may still be noisy or incomplete, we further propagate them across frames using SAM2 to obtain temporally consistent region tracks. This refinement serves two purposes: (i) smoothing out frame-to-frame inconsistencies of the heuristic cues, and (ii) propagating the initial proposals to frames both before and after the interaction moment, providing region coverage for the entire trajectory without requiring the heuristic to work perfectly on every frame. Therefore, the labeling process only requires coarse interaction-induced seeds, rather than precise frame-level annotations. More details are provided in Appendix A.6.

3.3 Guidance with Attention Weight Allocation

The heuristic regions C_{tar} and C_{rob} obtained in Section 3.2 provide cues about where the VLA should look. While these heuristic regions are not guaranteed to be perfectly accurate, they provide a useful attention prior that we exploit through soft constraints rather than hard labels. Here, we adopt a soft guidance strategy: rather than hard-constraining the attention distribution, we impose only a lower bound on the total attention allocated to the pseudo regions, encouraging VLAs to preserve focus on task-critical areas while retaining the freedom for action learning and remaining visual knowledge, without disrupting the visual understanding capabilities that make the VLM backbone valuable in the first place. Here, we use the attention map from the last layer of VLMs and average over all attention heads.

Language-to-image attention alignment. We first build the bridge between the semantics of the language instruction and the visual understanding with the guidance of C_{tar}. Concretely, let M^{l2i} denote the normalized attention weight matrix from the N_L language tokens to the image patch tokens. We first obtain the average attention weight matrix:

$$\overline{M}^{l2i}_{(i,j)} = \frac{1}{N_l} \sum_{m=1}^{N_l} M^{l2i}_{m,(i,j)}. \tag{5}$$

Then, we calculate how much weight is assigned to the region of the target item patches, and align the semantics of language and vision by encouraging enough attention weight for target items, which can be formulated as follows:

$$L_{l2i} = [max(0, \eta_1 - \sum_{(i,j) \in C_{tar}} \overline{M}^{l2i}_{(i,j)})]^2, \tag{6}$$

where η_1 denotes the minimum threshold for target item attention weight assignments, encouraging VLAs to maintain focus on the instruction-specified object without over-constraining action learning.

State-to-image attention alignment. While L_{l2i} connects language to the target objects, another alignment is needed between the robot state and the robot visual embodiment. Accurate action generation depends on the model's ability to relate its state to the spatial configuration of the robot in

the image. Using the candidate robot label C_{rob} and the normalized attention weight matrix M^{s2i} from the state token to the image patch tokens, we apply an analogous constraint:

$$L_{s2i} = [max(0, \eta_2 - \sum_{(i,j)\in C_{rob}} M^{s2i}_{(i,j)})]^2, \tag{7}$$

where η_2 denotes the minimum attention mass encouraged on candidate robot regions. Together, L_{l2i} and L_{s2i} provide explicit, region-specific guidance that directly counteracts the tendency of VLA attention to drift toward task-irrelevant regions during fine-tuning. By softly anchoring the model's attention to the regions that physically matter for the task, these objectives help the VLA retain focused visual perception throughout fine-tuning, directly addressing the attention drift problem identified in our motivating examples.

3.4 Guidance with Contrastive Learning

The guidance in Section 3.3 provides supervision at the level of attention weights, while explicitly shaping the semantic relationship between language instructions and the visual features extracted from those regions is also required. To strengthen this relationship, we turn to the representation space. Our key insight is that the regions C_{tar} from Section 3.2 naturally partition each pre-interaction frame into two semantically distinct sets: patches inside the refined candidate target region are treated as pseudo-positive visual evidence, while patches outside the target regions are treated as context negatives. The former should be semantically close to the language instruction, while the latter should not. Based on this, we design a contrastive learning objective that takes the language instruction representation as an anchor, pulls the visual features of C_{tar} toward it, and pushes the features of non-target regions away. This provides patch-level supervision that explicitly aligns the model's visual encoding of the target object with its linguistic understanding of the task. Formally, let P^{lan} denote the encoded representations of the N_l language instruction tokens, and P^{img} be the encoded representations of the image patch tokens from the VLA. We adopt the mean pooling strategy to obtain the representations of the semantic anchor, positive visual parts and negative visual parts:

$$\overline{P}_{lan} = \frac{1}{N_l}\sum_{m=1}^{N_l} P^{lan}_m, \overline{P}^{+}_{img} = \frac{1}{|C_{tar}|}\sum_{(i,j)\in C_{tar}} P^{img}_{(i,j)}, \overline{P}^{-}_{img} = \frac{1}{N_P - |C_{tar}|}\sum_{(i,j)\notin C_{tar}} P^{img}_{(i,j)}, \tag{8}$$

where N_P is a fixed number of image patches consistent with the image processing method of the VLM backbone. Then the formulation of contrastive learning is as follows:

$$L_{cl} = -\log \frac{\exp(sim(\overline{P}^{+}_{img}, \overline{P}_{lan}))}{\exp(sim(\overline{P}^{+}_{img}, \overline{P}_{lan})) + \exp(sim(\overline{P}^{-}_{img}, \overline{P}_{lan}))}, \tag{9}$$

where $sim(,)$ denotes the cosine similarity function. We apply L_{cl} only before interaction, where the target object is typically visible and less occluded by the robot, making language-target alignment cleaner.

3.5 VLA jointly training

With the procedures above, we have three training objectives for VLAs: L_{FM} for action generation learning, L_{l2i} and L_{s2i} for explicit attention weight allocation guidance, and L_{cl} for target-oriented representation enhancement. Together, these two forms of supervision address the perception gap from complementary angles: attention allocation shapes the model's visual focus on task-critical regions, while contrastive alignment ensures the visual features of those regions are semantically aligned with the language instruction. Finally, A^3-VLA is jointly optimized with these tasks with the following formulation:

$$L = L_{FM} + \lambda_1(L_{l2i} + L_{s2i}) + \lambda_2 L_{cl}, \tag{10}$$

where λ_1 and λ_2 are the weights of loss functions. It's notable that our framework is model-agnostic and can be seamlessly adapted to diverse VLAs. Notably, during inference, A3-VLA uses only the fine-tuned VLA policy, thus with no additional computational cost.

Table 1: Success Rate of VLAs on VLABench. The scores of the best baselines are underlined, while the scores of A^3-VLA are bold.

Method	Chemistry	Fruit	Poker	Toy	Mahjong	Avg.
Octo[27]	0.02	0.00	0.00	0.02	0.04	0.02
OpenVLA [16]	0.08	0.06	0.04	0.10	0.06	0.07
RDT [17]	0.12	0.16	0.08	0.10	0.06	0.10
π_0-FAST [20]	0.20	0.34	0.18	0.22	0.10	0.21
π_0 [4]	<u>0.52</u>	<u>0.60</u>	0.54	<u>0.40</u>	0.30	<u>0.47</u>
A^3-VLA (π_0)	**0.82**	**0.66**	**0.82**	**0.46**	**0.42**	**0.64**
OpenVLA-OFT [15]	0.32	0.18	0.14	0.12	0.04	0.16
A^3-VLA (OpenVLA-OFT)	**0.62**	**0.50**	**0.22**	**0.36**	**0.30**	**0.40**
$\pi_{0.5}$ [12]	0.30	0.42	<u>0.64</u>	0.32	<u>0.40</u>	0.41
A^3-VLA ($\pi_{0.5}$)	**0.74**	**0.50**	**1.00**	**0.42**	**0.68**	**0.67**

4 Experiment

In this section, we introduce details of the experiments on the evaluation of A^3-VLA. We first compare our method with state-of-the-art VLA baselines on both simulation and real-world environments. Then, the ablation study and training efficiency analysis are conducted to display the impact of the designed components. Last, we present the case study to demonstrate how perception guidance improves the performance of VLAs.

4.1 Experiment Setup

Simulation Benchmarks. For simulation, we evaluate our methods on VLABench [42] and Robotwin 2.0 [19, 7] benchmarks. VLABench provides a large scale of tasks to test the understanding of VLAs for semantic instructions and scene analysis, while Robotwin is a widely adopted benchmark with a large variety of scenes and objects for bimanual robotic simulation. For VLABench, we train and test models on five representative task suites: Chemistry Tube, Fruit, Poker, Toy and Mahjong, each containing 500 demonstrations. We test the VLAs over 50 episodes on each task suite in VLABench. For Robotwin, we compare the VLAs on four tasks: Beat Block Hammer, Click Bell, Move Playingcard Away, and Place Burger & Fries. For each task, we utilize Robotwin pipeline to generate 100 demonstrations for training, and test VLAs over 100 episodes. In each environment, tasks are all jointly trained.

Real-World Environment. In the real-world experiments, we set up scenes of manipulation tasks in the kitchen with a Songling robotic arm. The tasks include Open Cooker, Pick & Place Bowl, Pick 2 Vegetables, Select Bottles, and Desktop Sweep, each containing 80 demonstrations. Evaluation for each task is conducted 20 times, and we further record the task progress score, which is set as 1 if the task is completed, 0.5 if the VLA fails halfway, and 0 for inconsistency with instructions. In the real-world experiments, tasks are all trained together.

Implementation Details. We implement A^3-VLA on π_0, OpenVLA-OFT, and $\pi_{0.5}$, adopting each backbone's original training settings. For hyperparameters, the window size w is the same as the frame rate of videos. The variance threshold τ is set as 10, while the attention weight thresholds η_1 and η_2 are both 0.25. The loss weights λ_1 and λ_2 are set as 0.1 and 0.01 respectively. All the training and experiments are conducted on a server with 8 NVIDIA H200 GPUs.

Baselines. We compare A^3-VLA with various state-of-the-art methods. Octo [27] and OpenVLA [16] are pioneering works for open-source VLAs with OpenX pretraining. RDT [17] is a VLA with unified action space for transferable physical knowledge. π_0 [4] and π_0-FAST [20] are flow-based VLAs that complete manipulation tasks by action chunk generation. We also evaluate the top-performing OpenVLA-OFT [15] and $\pi_{0.5}$ [12]. OpenVLA-OFT is an improved version of OpenVLA which improves action learning by the optimized fine-tuning recipe. $\pi_{0.5}$ is an enhanced version of π_0 which incorporates open world knowledge from heterogeneous sources of data.

Table 2: Success Rate of VLAs on RoboTwin 2.0. The scores of the best baselines are underlined, while the scores of A^3-VLA are bold.

Method	Block Hammer	Click Bell	Playing Card	Burger & Fries	Avg.
OpenVLA	0.00	0.09	0.14	0.00	0.06
RDT	0.17	0.44	0.30	0.08	0.23
π_0	0.29	<u>0.56</u>	0.70	0.12	0.42
A^3-VLA (π_0)	**0.36**	**0.65**	**0.79**	**0.26**	**0.52**
OpenVLA-OFT	0.06	0.26	0.39	0.10	0.20
A^3-VLA (OpenVLA-OFT)	**0.08**	**0.37**	**0.46**	**0.18**	**0.27**
$\pi_{0.5}$	<u>0.56</u>	0.44	<u>0.81</u>	<u>0.52</u>	<u>0.58</u>
A^3-VLA ($\pi_{0.5}$)	**0.59**	**0.51**	**0.89**	**0.60**	**0.65**

Table 3: Success Rate (SR) and Task Progress (TP) of VLAs on Real-world Settings. Bold scores represent the highest results among all the methods, while the second highest scores are underlined.

	Open Cooker		Pick & Place Bowl		Pick 2 Vegetables		Select Bottles		Desktop Sweep		Avg.	
Method	SR	TP	SR	TP	SR	TP	SR	TP	SR	TP	SR	TP
π_0 [4]	<u>0.85</u>	<u>0.93</u>	<u>0.75</u>	<u>0.88</u>	0.40	0.58	0.25	0.30	0.50	0.65	0.57	0.68
$\pi_{0.5}$ [12]	**0.90**	**0.95**	0.65	0.83	<u>0.45</u>	<u>0.60</u>	<u>0.40</u>	<u>0.40</u>	<u>0.70</u>	<u>0.85</u>	<u>0.62</u>	<u>0.73</u>
A^3-VLA (π_0)	**0.90**	**0.95**	**0.80**	**0.90**	**0.55**	**0.78**	**0.75**	**0.80**	**0.80**	**0.90**	**0.76**	**0.87**

4.2 Comparison Results

VLABench Results. The results of experiments on VLABench are shown in Table 1. In general, the performance of A^3-VLA successfully surpasses all baselines. Compared to the backbones π_0, OpenVLA-OFT, and $\pi_{0.5}$, A^3-VLA gains average 16.4%, 24.0%, and 25.2% success rate improvements respectively, showing the consistent advancement of our method across different VLA frameworks. Specifically, A^3-VLA (π_0) improves higher in selecting Chemistry tube, Poker, and Mahjong compared to π_0, achieving 30%, 28%, and 12% of performance gains respectively. The improvement is particularly pronounced on tasks with visually similar distractors (e.g., Poker, Mahjong), demonstrating the effectiveness of our perception guidance for fine-grained object discrimination.

Robotwin Results. We evaluate our approach and other baselines on Robotwin 2.0, and report the quantitative results in Table 2. As shown in Table 2, A^3-VLA also improves all backbones on RoboTwin 2.0, with average gains of 9.8%, 7.0%, and 6.5% percentage points. The largest gains appear on Place Burger & Fries, suggesting that perception guidance is helpful for longer-horizon sequential manipulation.

Real-world Results. We make real-world experiments and compare the performance of our approach with π_0 and $\pi_{0.5}$ on various kitchen-themed manipulation tasks. The results are shown in Table 3, which demonstrates the advancement of A^3-VLA to complete different real-world tasks. Overall, our method outperforms the strongest baseline $\pi_{0.5}$ with 14% average success rates and 14% average task progress score. Among these tasks, A^3-VLA gains the most significant improvement on Select Bottles, which is a task demanding VLAs to pick up the specified bottle. This indicates that our design is helpful for VLAs' instruction understanding in the real-world background. By contrast, without perception-guided attention alignment, π_0 and $\pi_{0.5}$ have difficulty selecting the correct bottles. However in general, $\pi_{0.5}$ performs better than π_0 in various tasks.

4.3 Ablation Study

To verify the effectiveness of our designs, we conduct an ablation study on the key components of A^3-VLA. The results are displayed in Table 4, which demonstrates that our designs are consistently effective for robot manipulation. Among these training objectives, the improvement of L_{l2i} is the

Table 4: The ablation Study of A^3-VLA on VLABench. C/F/P/T/M denote five tasks of VLABench respectively.

Variations	C	F	P	T	M	Avg.
A^3-VLA (π_0)	**0.82**	**0.66**	**0.82**	**0.46**	**0.42**	**0.64**
w/o L_{l2i}	0.62	0.60	0.60	0.42	0.34	0.52
w/o L_{s2i}	0.76	0.62	0.78	0.46	0.38	0.60
w/o L_{cl}	0.78	0.66	0.80	0.40	0.40	0.61
π_0	0.52	0.60	0.54	0.40	0.30	0.47

Table 5: Evaluation under texture (TP) and Gaussian noise (NP) perturbations on VLABench.

Method	C	F	P	T	M	Avg.
$\pi_{0.5}$	**0.30**	**0.42**	**0.64**	**0.32**	**0.40**	**0.41**
+ TP	0.18	0.04	0.20	0.26	0.20	0.18
+ NP	0.26	0.24	0.60	0.26	0.30	0.33
A^3-VLA ($\pi_{0.5}$)	**0.74**	**0.50**	**1.00**	**0.42**	**0.68**	**0.67**
+ TP	0.60	0.10	0.66	0.30	0.46	0.42
+ NP	0.62	0.32	0.94	0.36	0.60	0.57

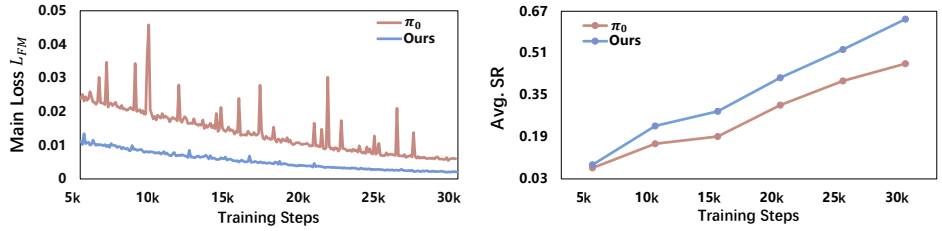


Figure 3: The comparison of L_{FM} and progressive SR of π_0 and A^3-VLA (π_0) on VLABench.

most significant, dropping 12% average success rate when removing it, since it enhances semantic understanding and visual analysis more directly by aligning the attention map of language querying images with target item patches. L_{cl} and L_{s2i} bring moderate advancements, dropping 3% and 4% average success rate respectively when blocking them, as they benefit representation learning and alignment between vision and robot states respectively. Overall, these auxiliary training tasks can jointly boost the performance and the capability of multimodal perception for VLAs.

4.4 Perturbation Robustness

For robustness, we test A^3-VLA towards two kinds of perturbations: 1) Texture perturbations (TP) test VLAs by changing the environments into unseen textures. 2) Noise perturbations (NP) test VLAs by adding Gaussian noise to the input images. We set the standard deviation of the Gaussian noise as $10/255$. The results are shown in Table 5. A^3-VLA remains stronger than $\pi_{0.5}$ under both perturbations, suggesting that the learned perception guidance improves robustness without using any external module at test time. Examples of this study are in Appendix A.2.

4.5 Training Efficiency

To validate the effect of A^3-VLA on the action learning progress, we illustrate the value of L_{FM} during training and evaluate their performance every 5k training steps for π_0 and A^3-VLA, which are shown in Figure 3. It shows a clear performance gap between A^3-VLA and the backbone π_0 under identical training settings. Specifically, A^3-VLA not only learns action policies more rapidly but also exhibits a smoother and more stable convergence curve, indicating a more robust learning process. This trend suggests that the auxiliary perceptual objectives may facilitate action learning by providing structured visual grounding. L_{l2i} encourages language-target association, L_{s2i} links robot states with visual embodiment, and L_{cl} strengthens language-target alignment, reducing the burden of finding task-relevant cues solely from the action loss.

4.6 Case Study

To achieve a deeper analysis of the effect of multimodal perception guidance, we further visualize the attention map $\overline{M}^{l2i}$ of A^3-VLA to investigate the alignment result for attention allocation guidance, and compare it with π_0 and $\pi_{0.5}$, illustrated in Figure 4. Through these cases, we can have the following observations. Firstly, for A^3-VLA, the regions with the highest attention weights largely correspond to the target objects in the language instructions. This demonstrates the effectiveness

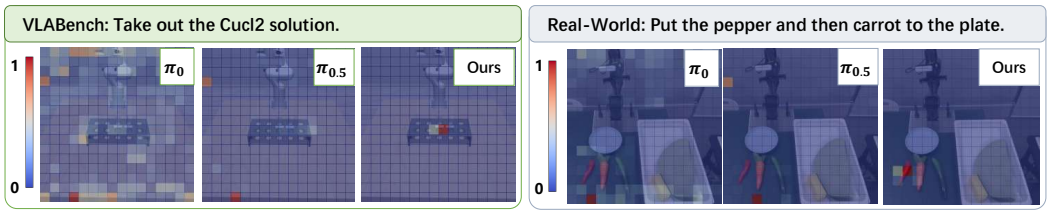


Figure 4: The visualization of the attention map $\overline{M}^{l2i}$ for language querying the image when testing π_0, $\pi_{0.5}$ and A^3-VLA (π_0) on different tasks and benchmarks.

of semantic alignment between language and visual observations with explicit target object patch guidance. Compared with π_0 and $\pi_{0.5}$, A^3-VLA concentrates attention more consistently around instruction-specified targets across both simulated and real-world tasks, supporting the effect of perception-guided attention alignment.

5 Conclusion

In this paper, we proposed A^3-VLA, a multi-modal perception-enhanced VLA framework that improved task-oriented visual understanding for robot manipulation. Specifically, to incorporate manipulation-progress information into language-visual understanding, A^3-VLA automatically generated heuristic candidate regions for target objects and the robot from manipulation trajectories, which were then temporally refined into pseudo regions using SAM2. To guide VLAs to allocate sufficient attention to task-critical pseudo regions and learn higher-quality semantic representations, we proposed two alignment training objectives: one encouraged attention weights to concentrate on candidate target or robot regions when processing language commands and observations, while the other used contrastive learning to bring target-region representations closer to instructions. Extensive experiments on both simulated and real-world environments validated the effectiveness of the proposed method.

6 Acknowledgments

This work was supported in part by New Generation Artificial Intelligence-National Science and Technology Major Project (Grant No. 2025ZD0122601), the National Natural Science Foundation of China (Grant No. 62506348), the Natural Science Foundation of Anhui Province (Grant No. 2508085QF211), and the Opening Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK (COGOS-2025HE02), the National Natural Science Foundation of China (Grant No.92370204), the National Key R&D Program of China (Grant No.2023YFF0725001), Guangdong Provincial Key Laboratory of Frontier Basic Science for All-domain Intelligence, the guangdong Basic and Applied Basic Research Foundation (Grant No.2023B1515120057), the Key-Area Special Project of Guangdong Provincial Ordinary Universities (2024ZDZX1007), and the National Natural Science Foundation of China (Grant No. 62506185).

7 Competing Interests

The authors declare no competing interests.

References

[1] Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, et al. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024.

[2] 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.

[3] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[6] 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.

[7] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[9] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[10] Shichao Fan, Kun Wu, Zhengping Che, Xinhua Wang, Di Wu, Fei Liao, Ning Liu, Yixue Zhang, Zhen Zhao, Zhiyuan Xu, et al. Xr-1: Towards versatile vision-language-action models via learning unified vision-motion representations. *arXiv preprint arXiv:2511.02776*, 2025.

[11] Ankit Goyal, Hugo Hadfield, Xuning Yang, Valts Blukis, and Fabio Ramos. Vla-0: Building state-of-the-art vlas with zero modification. *arXiv preprint arXiv:2510.13054*, 2025.

[12] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. π0. 5: a vision-language-action model with open-world generalization, 2025. *URL https://arxiv.org/abs/2504.16054*, 1(2):3, 2025.

[13] Nikita Kachaev, Mikhail Kolosov, Daniil Zelezetsky, Alexey K. Kovalev, and Aleksandr I. Panov. Don't blind your vla: Aligning visual representations for ood generalization, 2025.

[14] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838, 2022.

[15] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[16] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pages 2679–2713. PMLR, 2025.

[17] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[18] Junjie Meng, Ranxu Zhang, Zi-an Zhang, Shujun Liu, Xiaoning Qi, Xiaozhou Xu, Yanyong Zhang, Hui Xiong, and Chao Wang. Cedar: Controlled and event-driven demand forecasting via residual decomposition. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 7797–7808, 2026.

[19] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In *European Conference on Computer Vision*, pages 264–273. Springer, 2024.

[20] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[21] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[22] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

[23] 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.

[24] 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.

[25] Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*, 36:55681–55693, 2023.

[26] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[27] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[28] Chao Wang, Yixin Song, Jinhui Ye, Chuan Qin, Dazhong Shen, Lingfeng Liu, Xiang Wang, and Yanyong Zhang. Face: A general framework for mapping collaborative filtering embeddings into llm tokens. *Advances in Neural Information Processing Systems*, 38:146012–146039, 2026.

[29] Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. *arXiv preprint arXiv:2509.09372*, 2025.

[30] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[31] Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. Vlm: Task-agnostic video-language model pre-training for video understanding. *arXiv preprint arXiv:2105.09996*, 2021.

[32] Yifan Yang, Zhixiang Duan, Tianshi Xie, Fuyu Cao, Pinxi Shen, Peili Song, Piaopiao Jin, Guokang Sun, Shaoqing Xu, Yangwei You, et al. Fpc-vla: A vision-language-action framework with a supervisor for failure prediction and correction. *arXiv preprint arXiv:2509.04018*, 2025.

[33] Yunqian Yang, Qi Zhang, Shijin Wang, Yanyong Zhang, and Chao Wang. Cogen: A two-stage unified multi-modal framework for bundle construction via discrete semantics transfer. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6080–6091, 2026.

[34] Yan Zeng, Xinsong Zhang, Hang Li, Jiawei Wang, Jipeng Zhang, and Wangchunshu Zhou. X^{2}-vlm: All-in-one pre-trained model for vision-language tasks. *IEEE transactions on pattern analysis and machine intelligence*, 46(5):3156–3168, 2023.

[35] Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Fuxian Huang, Haoran Zhang, Ming Zhou, Shengzhe Zhang, Litao Liu, Sixu Lin, and Jiangmiao Pang. A vision-language-action-critic model for robotic real-world reinforcement learning, 2025.

[36] Zhihao Zhan, Jiaying Zhou, Likui Zhang, Qinhan Lv, Hao Liu, Jusheng Zhang, Weizheng Li, Ziliang Chen, Tianshui Chen, Ruifeng Zhai, et al. E0: Enhancing generalization and fine-grained control in vla models via tweedie discrete diffusion. *arXiv preprint arXiv:2511.21542*, 2025.

[37] 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, 2026.

[38] 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.

[39] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*, 2025.

[40] Shengzhe Zhang, Liyi Chen, Dazhong Shen, Chao Wang, and Hui Xiong. Hierarchical time-aware mixture of experts for multi-modal sequential recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 3672–3682, 2025.

[41] Shengzhe Zhang, Liyi Chen, Chao Wang, Shuangli Li, and Hui Xiong. Temporal graph contrastive learning for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 9359–9367, 2024.

[42] Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11142–11152, 2025.

[43] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.

[44] Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Yaxin Peng, Chaomin Shen, Feifei Feng, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5377–5395, 2025.

[45] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

A Appendix

A.1 Parameter Sensitivity

In this part, we study the impact of important hyper-parameters of A^3-VLA, including the loss weight of the guidance with attention weight allocation λ_1, the loss weight of contrastive learning λ_2, and the threshold of attention weight allocation for language querying images η_1. When changing one hyper-parameter, we keep other hyper-parameters fixed to control variables. The results are illustrated in Figure 5. First, for λ_1, we can see that a lighter weight for the guidance with attention weight allocation is sufficient to improve the performance significantly. Whereas as the weight increases, the performance of A^3-VLA drops, probably because A^3-VLA gets too stuck to attention weight allocation and overlooks the action learning from L_{FM}. λ_2 shows the similar tendency as λ_1. Next, for η_1, the results demonstrate that the guidance with attention weight allocation is the most powerful for the VLA enhancement when η_1 has a moderate value. As it gets larger, the VLA may pay too much attention to the target regions while ignoring whole scene information, slightly weakening the improvement from the guidance with attention weight allocation.

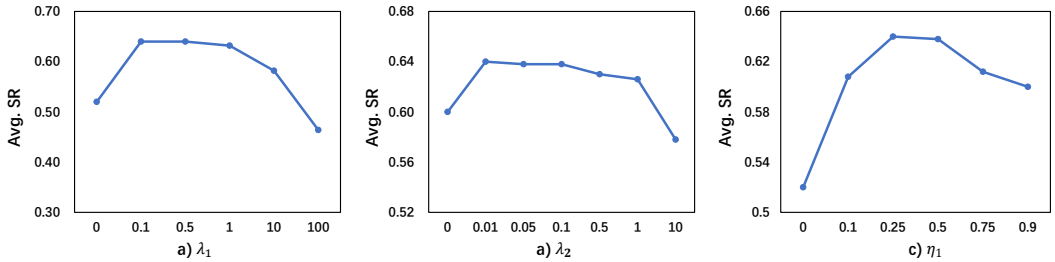


Figure 5: Performance comparison on VLABench of A^3-VLA w.r.t different hyper-parameters.

A.2 Examples for Perturbation Robustness

We visualize examples for the manipulation process of A^3-VLA ($\pi_{0.5}$) and $\pi_{0.5}$ under texture and noise perturbations in Figure 6. We can see that A^3-VLA ($\pi_{0.5}$) still performs better, showing our robustness to perturbations such as background texture changing.

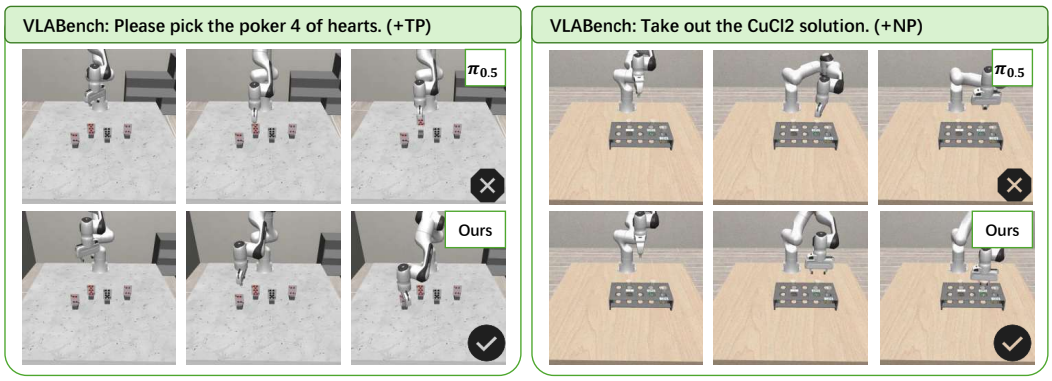


Figure 6: The manipulation process of A^3-VLA ($\pi_{0.5}$) and $\pi_{0.5}$ under perturbations on VLABench.

A.3 More Examples for Case Study

We present more examples of the case study in Figure 7. The conclusion is consistent: the regions with the highest attention weights of A^3-VLA largely correspond to the target objects specified by the language instructions. The attention allocation of π_0 is quite blurry, which is scattered throughout the whole image. Whereas $\pi_{0.5}$ sticks to specific positions of patches. In the last case, it's observed

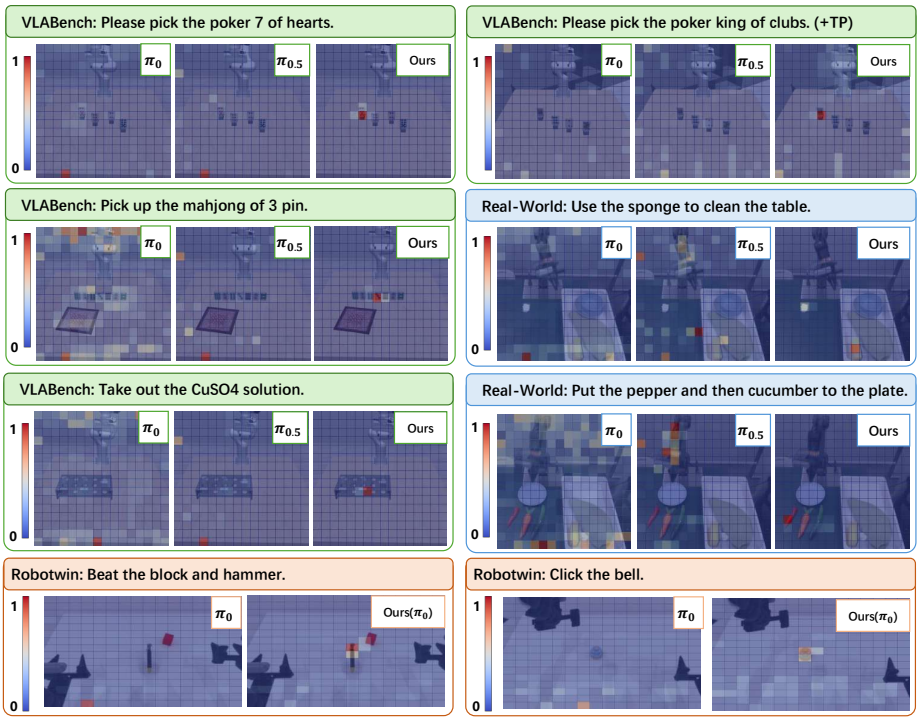


Figure 7: The visualization of the attention map $\overline{M}^{l2i}$ for language querying the image when testing π_0, $\pi_{0.5}$ and A^3-VLA (π_0) on different tasks and benchmarks.

that the attention except for the pepper shifts from the carrot to the cucumber compared with the last case in Figure 4, showing the semantic understanding of A^3-VLA. Further more, under the unseen texture perturbations (TP), A^3-VLA is still able to find out the regions of the target object, showing our robustness.

A.4 More Examples for Heuristic Region Proposal

In this section, we show more examples of the results from the heuristic region process in Figure 8 and Figure 9. The first task is to put the spiderman into the gift box, while the second task is to click the bell. Though in different benchmarks, our method can localize candidate regions of target items and robots across different situations and environments. This indicates that our method is useful under different situations and environments.

A.5 More Details of Implementation

The training batch sizes of π_0 and $\pi_{0.5}$ are set as 32 and 256 respectively, and they are both trained in 30k steps. For OpenVLA-OFT, we set the batch size and training steps as 8 per GPU and 50k, respectively. For the choice of τ, we find that $\tau = 10$ works well across datasets, and slight changes to this value do not noticeably affect the results. Therefore, the heuristic region proposal is not sensitive to this threshold. All baselines are fine-tuned with the dataset of the benchmarks.

A.6 Details of SAM2-based Temporal Refinement

We use SAM2 as an off-the-shelf temporal refinement tool to improve the consistency of the heuristic region proposals. Given the candidate target and robot regions obtained from interaction-induced visual changes, we first convert the selected image patches at the interaction frame into binary masks. These binary masks are directly used as mask prompts for SAM2. Target and robot regions are prompted and refined separately using their corresponding heuristic masks. Starting from the prompted interaction frame, SAM2 propagates the masks along the demonstration trajectory to

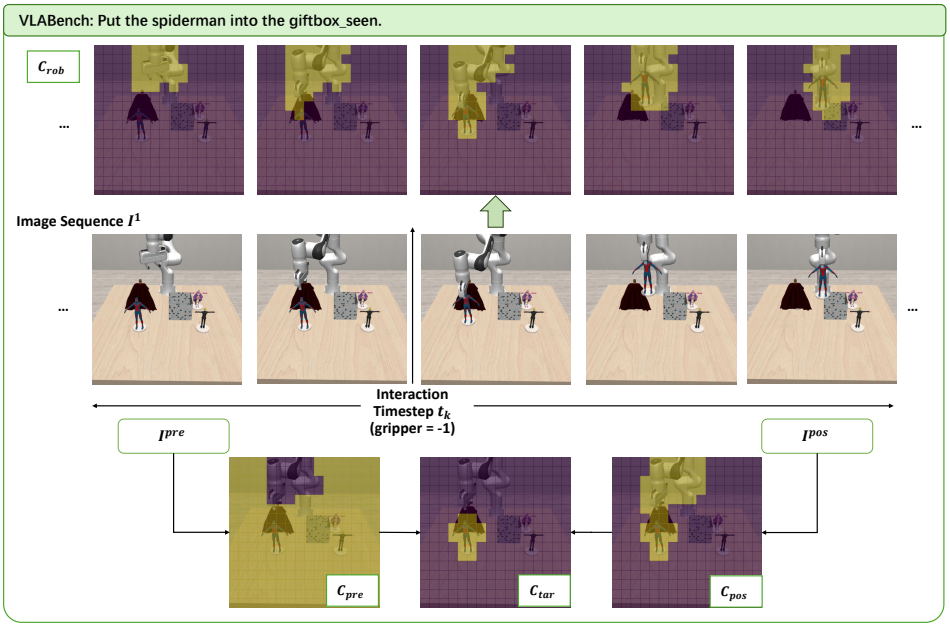


Figure 8: An Example of heuristic region proposals on VLABench.

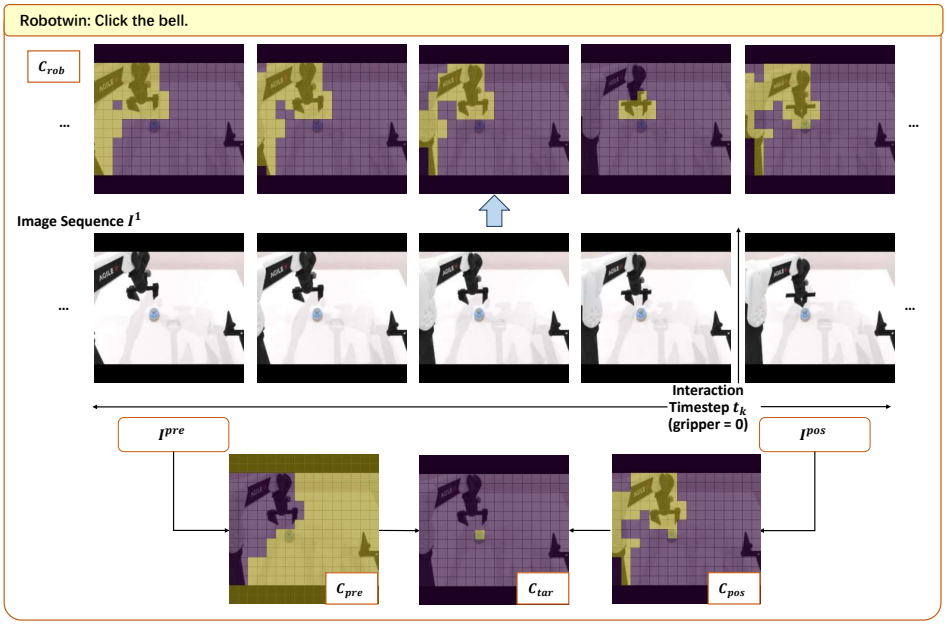


Figure 9: An Example of heuristic region proposals on Robotwin.

produce temporally consistent pseudo masks for each frame. This refinement helps reduce isolated noisy patches and provides region estimates for frames where the original heuristic cue is weak or unavailable. The resulting masks are not treated as ground-truth annotations; they are used only as pseudo regions for soft perceptual supervision during VLA fine-tuning. We further map each refined pixel-level mask to the VLA patch grid. Let M_{tar}^t and M_{rob}^t denote the refined target and robot masks at timestep t, and let Ω denote the set of image patches. A patch is assigned to the target or robot pseudo region if its overlap ratio with the corresponding mask exceeds a threshold ρ:

$$C_{tar}^t = \left\{ p \in \Omega \mid \frac{|p \cap M_{tar}^t|}{|p|} > \rho \right\}, \tag{11}$$

$$C_{rob}^t = \left\{ p \in \Omega \mid \frac{|p \cap M_{rob}^t|}{|p|} > \rho \right\}. \tag{12}$$

The patch-level pseudo regions C_{tar}^t and C_{rob}^t are then used in the attention allocation losses and the contrastive objective. All pseudo masks are precomputed and cached before policy training. Therefore, SAM2 introduces additional offline preprocessing cost but no extra computation during policy inference. ρ is empirically set as 0.25.

A.7 Limitations

A^3-VLA relies on interaction-induced visual changes to obtain heuristic candidate regions. Although these regions are further refined by SAM2 and used only as soft supervision, the quality of the pseudo regions may degrade when the target object is fully occluded, visually indistinguishable from the background, too small to produce reliable patch-level changes, or moves together with distractors. In such cases, the auxiliary perception losses may provide weaker or noisier guidance.

Another limitation is that our current implementation assumes that a coarse interaction moment can be obtained from lightweight robot states or task-state cues, such as gripper states, contact signals, or predefined state ranges. While such cues are commonly available in manipulation demonstrations, more complex embodiments or tasks without clear interaction events may require more robust interaction discovery mechanisms.

A^3-VLA also introduces an offline preprocessing step using SAM2 to improve the temporal consistency of pseudo regions. This does not add inference-time cost, since SAM2 is not used during policy execution, but it increases data preprocessing time and introduces a dependence on the quality of the off-the-shelf segmentation model. Future work may explore uncertainty-aware weighting of pseudo regions, more robust interaction-guided region discovery, and fully end-to-end perception guidance without external refinement tools.

A.8 Broader Impact

Positive impacts include improving robot instruction following and reducing manual annotation for robot learning. Potential risks include failures in real-world deployment due to incorrect perception or attention alignment, which may cause unsafe robot actions. A^3-VLA should therefore be deployed with safety constraints, human supervision, and task-specific validation in physical environments.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly summarize our contributions and scope. We focus on VLAs and propose a framework to protect the vision ability of their VLM backbone.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the Appendix. By explicitly acknowledging these limitations, we provide a balanced view of our work and suggest directions for future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our method has no theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the details of benchmarks, implementation, hyperparameter settings, and pseudo-region generation in the paper and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We give the URL of our code. The datasets for VLABench and Robotwin are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details in the paper and the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Each task is evaluated dozens of times, and we report the average success rate for consistency and reliability.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information on compute resources, including GPUs, is provided in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms with NeurIPS Code of Ethics in every aspect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts of our works in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: In our experiments, we employ publicly available embodiment datasets that are widely used in the research community, together with established open-source VLMs that naturally comply with open-science principles. Because these datasets are non-sensitive and the underlying methods are fully transparent, we consider the risk of misuse to be negligible.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We adhere rigorously to these standards by guaranteeing that all assets used in our research—including code, datasets, and pre-trained models—are properly credited via references to their original authors.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We offer our code in an anonymous GitHub repository.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our research contains neither crowdsourcing experiments nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our research contains neither crowdsourcing experiments nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: VLMs are utilized as a backbone for VLAs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.