Affordance Field Intervention: Enabling VLAs to Escape Memory Traps in Robotic Manipulation

Supplementary Material

A. Implementation Details

Hardware Configuration. Our real-world experiment setup employs an *AgileX Piper manipulator* equipped with two *Intel RealSense D435 cameras*: one mounted on the wrist and another positioned in front of the robot. Both cameras are calibrated relative to the robot base frame to enable accurate 3D point cloud reconstruction. We utilize the calibrated front-mounted RealSense camera for scene reconstruction, operating at 30 Hz observation frequency. During data collection, we utilize RGB images from both viewpoints, while depth information is exclusively used at inference time to construct 3D point clouds for spatial affordance field generation.

Spatial Affordance Field Construction. For affordance field construction, we apply the GPT-4o API to parse task instructions and identify manipulation stages. Open-vocabulary detection and target object tracking are performed locally on an NVIDIA GeForce GTX 1080Ti GPU using Grounded-SAM [33] to generate 2D instance segmentation masks. The complete SAF is published and updated as a ROS topic at 2 Hz frequency, ensuring real-time spatial reasoning without introducing latency to the original VLA inference pipeline.

Control and Kinematics. Additionally, we deploy Curobo on the same 1080Ti GPU for forward and inverse kinematics computation. This enables bidirectional transformation: converting VLA-predicted joint states to end-effector spatial coordinates, and mapping SAF-sampled waypoints back to joint configurations. The kinematics computation introduces approximately 5ms latency and operates as a ROS service at 10Hz frequency. Overall, our SAF updates at 2Hz without adding overhead to the VLA policy inference pipeline, maintaining efficient real-time control.

Training Details. For data collection, we use a master-follower teleoperation setup with an auxiliary AgileX Piper arm. The baseline VLA models (π_0 and $\pi_{0.5}$) are fully fine-tuned on collected demonstration trajectories for 30,000 steps with batch size 32 on a single NVIDIA H100 GPU. During inference, we sample 8 action chunks per query to ensure trajectory diversity. For stochastic action generation via flow matching, we set the initial noise sampling temperature (standard deviation) to 1.5 to encourage diverse proposals, which are then re-ranked by our SAF-based scorer.

B. Real-world Task Settings

We evaluate our framework on four real-world manipulation tasks with varying complexity, each designed to test different manipulation primitives and robustness to distribution shifts. Each task is evaluated over 20 trials under five test conditions: in-distribution (ID) and four OOD scenarios (illustrated in Figure 7).

B.1. Task Description and Collection

Task 1 - Place Carrot. The instruction is "Pick up the carrot and place it in the pot." The objective is to pick up a carrot from a plate and place it into a pot. Success is defined as successfully placing the carrot into the pot. We collect 68 expert demonstration trajectories using a master-follower teleoperation setup.

Task 2 - Remove Lid. The instruction is "Remove the lid from the pot." The objective is to remove the stainless steel lid from a pot and place it onto a nearby platter. Success is defined as placing the lid completely within the platter boundaries. We collect 78 expert demonstration trajectories with the pot position fixed. The platter is divided into two regions (A and B), and the lid placement is distributed across both regions, with 39 trajectories collected for each region.

Task 3 - Slot Pen. The instruction is "Slot the yellow pen into the holder." The task requires inserting a yellow marker pen into a pen holder. Success is defined as the pen being fully inserted into the holder. We collect 77 expert demonstration trajectories for this task.

Task 4 - Stack Tape. The instruction is "Stack the brown tape on top of the grey tape." This task involves placing a roll of brown tape on top of a roll of grey tape. Success is defined as the brown tape resting stably on the grey tape without falling. We collect 80 expert demonstration trajectories for this task.

Task 5 - Remove Lid and Place Carrot (Long-Horizon). The instruction is "Remove the pot lid and place the carrot inside the pot." This is a long-horizon task comprising two sequential stages: Stage 1 requires removing the stainless steel lid from the pot and placing it on a nearby platter, and Stage 2 requires picking up a carrot and placing it into the pot. Success is defined as completing both stages in sequence. We collect 78 expert demonstration trajectories for this task.

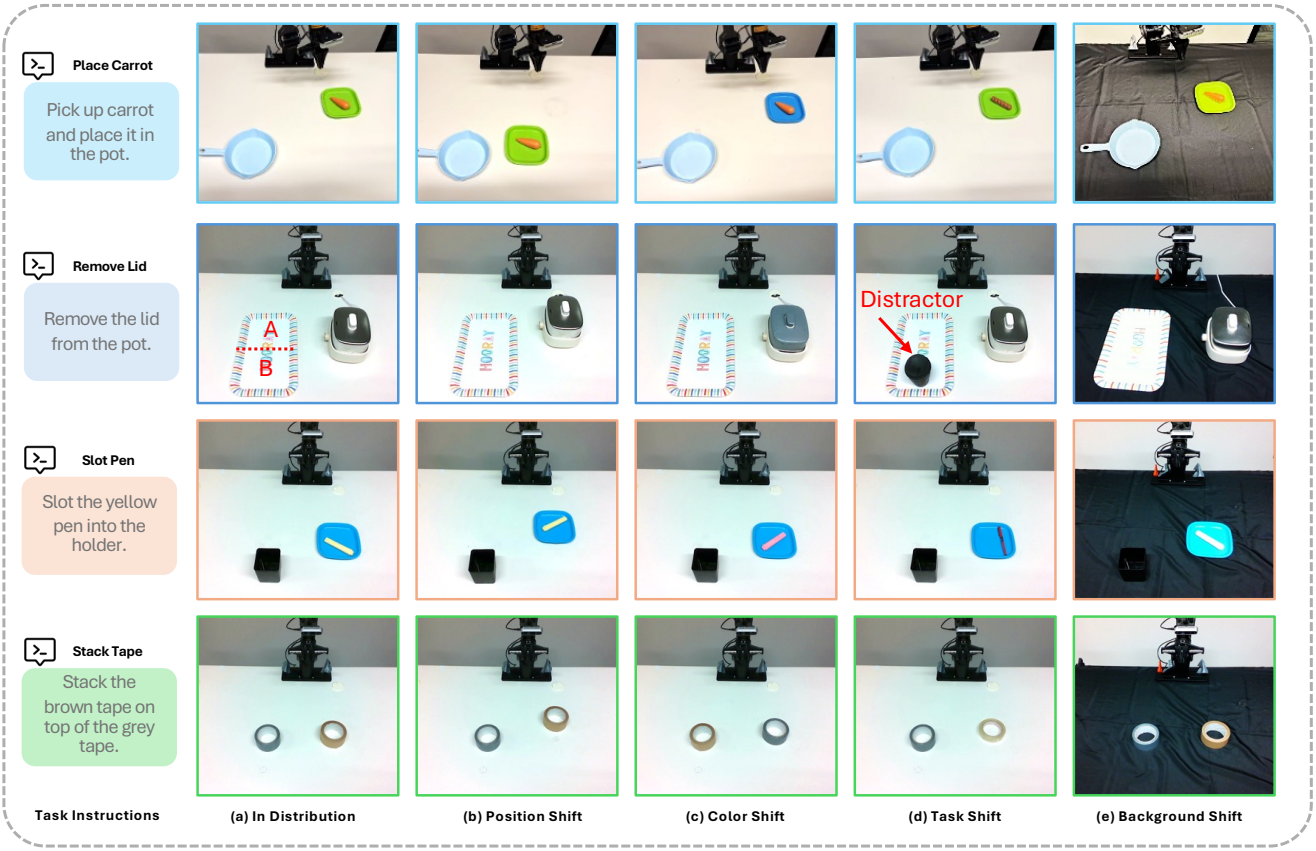

Figure 7. **Visual illustration of OOD test scenarios.** Four manipulation tasks across five test conditions: (a) in-distribution setting, (b) position shift (objects displaced by ±5-15cm), (c) color shift (object appearance change), (d) task shift (physical property variations or distractors), and (e) background shift (table surface color change from white to black). Each row shows a different task: Place Carrot (top), Remove Lid (second), Slot Pen (third), and Stack Tape (bottom).

B.2. Out-of-Distribution Test Scenarios

Following the evaluation protocol in our main experiments, we design four types of distribution shifts for each task. Figure 7 provides a visual illustration of these OOD test scenarios across all four tasks:

Position Shift. Objects are displaced within a 5-15cm radius from their training positions. For Place Carrot, the carrot is displaced by approximately 15 cm; for Remove Lid, we move the pot within 5cm; for Slot Pen, we move the blue plate containing the pen within 5cm; for Stack Tape, we move the tape placement area within 5cm.

Color Shift. For Place Carrot, the plate holding the carrot is changed from green to blue. For Remove Lid, the stainless steel lid is replaced with a grey-colored lid. For Slot Pen, the yellow marker is replaced with a pink one. For Stack Tape, the brown tape is replaced with a grey-colored tape.

Task Shift. For Place Carrot, the target object is replaced from carrot to sausage while maintaining the same task structure. For Remove Lid, we add a black cup as a distractor in the platter region and extend the instruction with the phrase "Avoid the black cup." For Slot Pen, the yellow marker is replaced with a thinner red pen, requiring different insertion dynamics. For Stack Tape, the brown tape is replaced with a different type of tape (different shape and texture).

Background Shift. The table surface color is changed from white to black across all tasks.

Clutter. For the Long-horizon task (*Remove Lid and Place Carrot*), we introduce 3–5 distractor objects randomly onto the table workspace to create visually dense scenes (Figure 8). The distractors increase perceptual complexity and the risk that the VLA policy will attend to irrelevant objects.

C. Supplementary Experiments

C.1. Memory Trap Detector Robustness

Threshold Sensitivity and Precision/Recall. We analyze the reliability of our dual-criterion detector (Sec. 4) across 160 trials. As shown in Table 5, the success rate is robust to a wide range of ϵ_{stuck} and ϵ_{far} values: SR remains at 80% for $\epsilon_{\text{stuck}} \in [2.0, 3.5]$ cm and SR $\geq 70\%$ for $\epsilon_{\text{far}} \in [22, 28]$ cm. Only extreme values outside these ranges cause noticeable

degradation, confirming that the thresholds do not require fine-tuning. Both thresholds are set from task geometry: ϵ_{stuck}=2.5 cm matches the EEF displacement statistics during fine manipulation (middle Figure 3), and ϵ_{far}=25 cm is set slightly above the 20 cm object placement area diameter. On a separate 40-trial FP/FN analysis, the detector achieves *Precision = 95%* (19/20) and *Recall = 90.5%* (19/21), with only one false positive and two false negatives.

Robustness to Depth and Calibration Noise. We test AFI under worst-case sensor conditions: depth noise σ=2 cm and camera calibration error δ=2 cm, applied independently to the *Place Carrot* task. The success rate remains at 60% under both conditions (0% degradation relative to clean sensors). This robustness arises from two properties of our SAF construction: (1) target centroid averaging over hundreds to thousands of point cloud samples reduces 2 cm depth noise by approximately $30\times$ ($\sigma/\sqrt{N}$); (2) the detection threshold ϵ_{far}=25 cm provides a large margin that is an order of magnitude larger than the 2 cm calibration error.

Table 5. Threshold sensitivity of memory trap detector (Place Carrot, position shift). SR: success rate. Thresholds are in cm.

ϵ_{stuck}	SR	ϵ_{far}	SR
1.5	55%	18	50%
2.0	80%	22	70%
2.5	**80%**	**25**	**80%**
3.0	80%	28	75%
3.5	80%	32	60%
4.0	65%	35	50%

Table 6. Ablation study on position shifts. We evaluate π_0 and π_0-AFI under different spatial displacements along X and Y axes (in centimeters).

$(\Delta X, \Delta Y)$	(0,0)	(+10,0)	(+15,0)	(0,+10)	(0,+15)	(+10,+10)
π_0	17/20	3/20	1/20	4/20	0/20	6/20
π_0-AFI	**20/20**	**8/20**	**3/20**	**10/20**	**2/20**	**13/20**

Robustness to position shifts along different axes. To investigate VLA models' robustness to spatial perturbations, we systematically shift the target object along X-axis and Y-axis independently and jointly. Table 6 presents results across six position configurations. π_0's performance degrades catastrophically under single-axis shifts: success rates drop to 15% for $\Delta X = +10$ cm and 5% for $\Delta X = +15$ cm, revealing that VLA models memorize specific spatial patterns and fail when deviating along unseen directions. Interestingly, diagonal displacement ($\Delta X = +10$ cm, $\Delta Y = +10$ cm) maintains 30% success, likely because such trajectories align with the training distribution's spatial coverage. Our method consistently improves robustness across all configurations (40% for single-axis, 65% for diagonal), validating that explicit 3D spatial guidance helps escape memory traps. However, extreme OOD scenarios ($\Delta X/Y = +15$ cm) show diminishing returns, highlighting that our approach complements rather than replaces the VLA's learned priors.

C.2. Additional VLA Backbone Results

To verify that AFI is truly backbone-agnostic, we evaluate on two additional VLA architectures beyond the π family: OpenVLA-OFT [22] (autoregressive token prediction) and SpatialVLA [32] (3D-aware backbone that encodes depth features). Table 7 reports results on the *Place Carrot* task under three primary OOD conditions.

OpenVLA-OFT achieves strong in-distribution performance (90%) but degrades sharply under position shift (15%) and background shift (40%). AFI recovers these to 35% and 70% respectively, yielding a +20% average improvement. This confirms that autoregressive VLAs exhibit the same memory trap phenomenon as flow-matching models, and that our spatial affordance guidance generalizes across decoding paradigms.

Notably, SpatialVLA—despite being explicitly trained with 3D depth inputs—still suffers severe memory traps under position shift (75%→30%), revealing that depth-aware training alone is insufficient to escape memorized spatial patterns under distribution shifts. AFI improves SpatialVLA by +25% on average (from 53.3% to 78.3%), with particularly large gains in background shift (55%→85%). This demonstrates that our explicit, test-time spatial affordance guidance provides complementary value even for architectures that already incorporate 3D representations during training.

Table 7. AFI on additional VLA backbones (Place Carrot task). I.D.=In-Distribution, Pos.=Position shift, B.G.=Background shift.

Method	I.D.	Pos.	B.G.	Avg
SpatialVLA	75%	30%	55%	53.3%
+AFI (Ours)	**95%**	**55%**	**85%**	**78.3%** (↑25%)
OpenVLA-OFT	90%	15%	40%	48.3%
+AFI (Ours)	**100%**	**35%**	**70%**	**68.3%** (↑20%)

C.3. Long-Horizon and Cluttered Scene Results

To evaluate AFI beyond single-stage pick-and-place, we conduct experiments on a long-horizon task: *"Remove the pot lid and place the carrot inside the pot."* This task comprises two sequential stages: lid removal (Stage 1) and carrot placement (Stage 2), and requires the SAF to update its semantic focus mid-execution. AFI handles this by constructing separate SAFs for each stage, activating the lid-SAF during Stage 1 and switching to the carrot-SAF after the lid is removed. We also introduce a *Clutter* condition (3–5 distractors on the table) to evaluate robustness to denser scenes.

Table 8 and Figure 8 summarize the results. AFI achieves +20% average improvement for π_0 (from 43.8% to 63.8%) and +16.2% for $\pi_{0.5}$ (from 43.8% to 60.0%), demonstrating effective multi-stage execution. The largest gains appear in position shift (+25% for π_0) and background

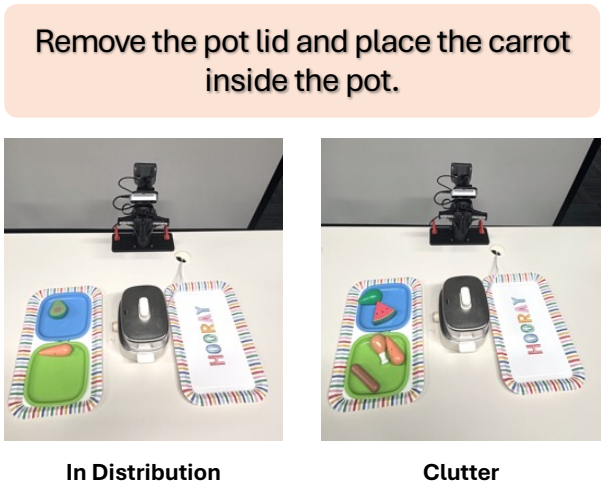

Figure 8. **Long-Horizon and Cluttered Scenes.** Left: in-distribution scene with only task-relevant objects. Right: Clutter condition with 3–5 distractors (e.g., toy watermelon, sausages) added to the workspace, increasing perceptual complexity.

shift (+25% for π_0), where spatial memorization is most problematic across sequential stages.

In the Clutter condition, baseline performance drops substantially (π_0: 20%, $\pi_{0.5}$: 10%), as distractors increase perceptual complexity and the risk of the VLA attending to irrelevant objects. AFI improves π_0 from 20% to 45% and $\pi_{0.5}$ from 10% to 35%, both gaining +25%. This validates that our language-conditioned SAF construction via Grounded-SAM correctly identifies the target object even in cluttered scenes, providing reliable spatial guidance despite visual distractors.

Table 8. Long-horizon task (*Remove Lid and Place Carrot*). Numbers show success rates (%) with improvement over backbone in parentheses. Clutter: 3–5 distractors added to the table.

Method	I.D.	Pos.	B.G.	Clutter	Avg
π_0	90	25	40	20	43.8
π_0-AFI (Ours)	**95** (+5)	**50** (+25)	**65** (+25)	**45** (+25)	**63.8** (+20)
$\pi_{0.5}$	90	30	45	10	43.8
$\pi_{0.5}$-AFI (Ours)	**100** (+10)	**45** (+15)	**60** (+15)	**35** (+25)	**60.0** (+16.2)

C.4. AFI Failure Mode Analysis

Based on over 200 real-world trials, we analyze the root causes of AFI failures as follows:

- **VLA policy errors (60%):** The underlying VLA generates physically incorrect actions (e.g., wrong grasp angle) even after being guided to the correct spatial location. These failures are orthogonal to SAF and reflect inherent VLA limitations.
- **Physical uncertainties (25%):** Unexpected contact dynamics (object slipping, table friction variations) cause execution failures after a successfully planned trajectory.
- **Segmentation errors (10%):** Grounded-SAM fails to correctly segment the target object (e.g., under severe lighting changes or heavy occlusion), leading to an incorrect SAF target region.
- **SAF guidance errors (5%):** SAF directs the robot to a geometrically reachable but task-suboptimal region. Importantly, we observed **no cases** where SAF guided the robot to a semantically incorrect region (e.g., wrong object), as Grounded-SAM's language-conditioned segmentation ensures semantic correctness.

The 5% SAF guidance error rate confirms that the affordance field itself is reliable in practice, and the dominant failure modes are shared with standalone VLA baselines (policy errors, physical uncertainties).

D. Discussion and Limitations

Remark on SAF as a Positional Module. SAF is intentionally designed as a *positional* guidance module: it corrects *where* the robot should move to escape a memory trap, while grasp orientation, contact force, and manipulation semantics are delegated to the frozen VLA policy. This design is motivated by our observation that memory traps predominantly manifest as end-effector mis-localization in 3D space rather than incorrect grasp kinematics. Consequently, pure spatial correction is sufficient for resolving the failure mode without requiring SAF to encode task-specific contact models.

Rollback Safety. The rollback trajectory retraces a recently visited path in the workspace. In the static and semi-static environments targeted by this work, this path is inherently collision-free, as it was traversed successfully by the robot moments before the memory trap was detected. Handling dynamic environments with obstacles that may move into the rollback path is left as future work.

Limitations. The current SAF is a *positional* module and does not encode grasp orientation, contact force, or tool-use semantics; tasks requiring precise approach angles (e.g., plug insertion) may benefit from extending SAF to SE(3) pose guidance, which we leave as future work. The rollback mechanism assumes a static or semi-static workspace, and safety guarantees in highly dynamic environments would require explicit collision checking along the replay path. Additionally, our current deployment runs SAF updates at 2 Hz via asynchronous ROS topics; higher-frequency dynamic tasks may necessitate faster local VLM inference.