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

Siyu, Xu[1]; Zijian Wang[1]; Yunke Wang[1], Chenghao Xia[3], Tao Huang[2], Chang Xu[1]
[1]The University of Sydney, [2]Shanghai Jiao Tong University, [3]StellarEdge Robotics

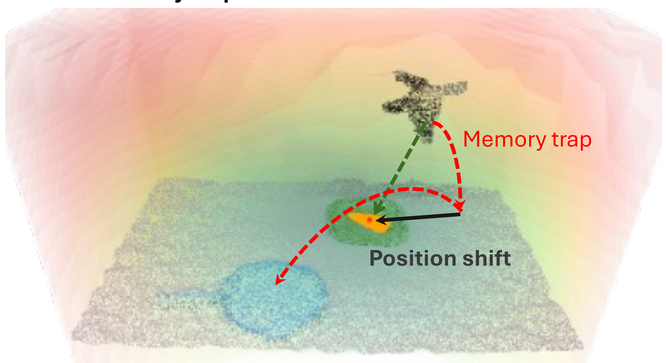
Contact: s.xu@sydney.edu.au

CVPR JUNE 3-7, 2026 DENVER COLORADO

Introduction

Memory Trap in VLAs

Under distribution shifts, VLAs **replay memorized trajectories** and reach the old target instead of the new one — a failure we call the **Memory Trap**.



Motivation

Why current approaches fail?

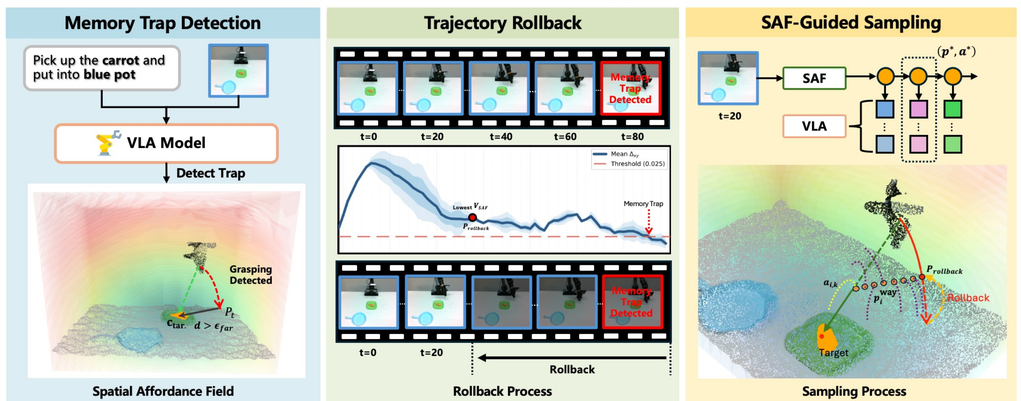
- **End-to-end VLAs** (OpenVLA, π_0) lack explicit 3D perception, can't locate actionable regions in *unfamiliar scenes*.
- **VLM-based planners** (VoxPoser, ReKep): produce geometrically infeasible actions and rely on *brittle, task-specific* prompts that don't transfer.

Key Insight

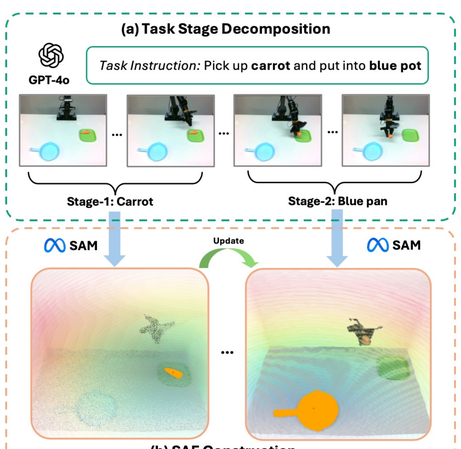
VLM grounds *where* to act; VLA decides *how* to act. AFI lets the 3D affordance field guide the VLA *on demand*, ***without parameter updates***.

Methods

AFI Framework



SAF Construction



4-step Pipeline

1. ***Memory Trap Detection:*** Proprioception flags a trap when the end-effector is **stuck yet far from the goal**: $\|\mathbf{p}_t - \mathbf{p}_{t-\Delta t}\| < \epsilon_{\text{stuck}}$ and $\|\mathbf{p}_t - \mathbf{c}_{\text{target}}\| > \epsilon_{\text{far}}$. Dual criteria avoid false alarms during normal grasping.
2. ***Historical Rollback:*** Roll back to the recent state with the **lowest SAF cost**, giving a safe, high-affordance restart point.
3. ***SAF-Guided Waypoint Sampling:*** Sample **N** high-affordance waypoints nearby: close to the goal, away from obstacles.
4. ***Trajectory Scoring & Execution:*** The VLA proposes several candidates per waypoint; pick the trajectory with the **highest cumulative affordance** and execute it.

Experiments

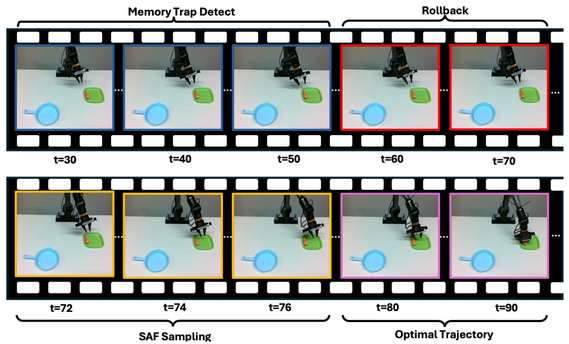
Quantitative results

Task	Method	In Dist.	Position	Color	Task	Background	Average SR.
Place Carrot	ReKep [13]	8/20	7/20	9/20	5/20	7/20	36.0%
	π_0	17/20	6/20	13/20	15/20	10/20	61.0%
	π_0-AFI (Ours)	**20/20**	**13/20**	**17/20**	**18/20**	**19/20**	**87.0%** (↑26.0%)
Remove Lid	π_0	20/20	8/20	17/20	5/20	13/20	63.0%
	π_0-AFI (Ours)	**20/20**	**12/20**	**19/20**	**11/20**	**18/20**	**80.0%** (↑17.0%)
Slot Pen	π_0	16/20	11/20	13/20	15/20	5/20	60.0%
	π_0-AFI (Ours)	**19/20**	**16/20**	**16/20**	**19/20**	**12/20**	**82.0%** (↑22.0%)
Stack Tape	π_0	18/20	9/20	16/20	13/20	8/20	64.0%
	π_0-AFI (Ours)	**20/20**	**15/20**	**20/20**	**17/20**	**14/20**	**86.0%** (↑22.0%)
	$\pi_{0.5}$	20/20	7/20	17/20	10/20	7/20	61.0%
	$\pi_{0.5}$-AFI (Ours)	**20/20**	**14/20**	**19/20**	**15/20**	**14/20**	**82.0%** (↑21.0%)
	π_0+$\pi_{0.5}$-AFI (Ours)	**20/20**	**16/20**	**20/20**	**16/20**	**17/20**	**89.0%** (↑25.0%)

Consistent **+17–26%** gains across all tasks and backbones, largest under the hardest shifts. *Ensemble* reaches **89%**; ReKep only 36%.

Qualitative results

One rollout shows the full loop:



Generalization across backbones

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%)

Backbone-agnostic: +20% on *OpenVLA-OFT* and **+25%** on *SpatialVLA*. Even depth-aware training doesn't prevent memory traps, so explicit affordance guidance is **complementary**.