RAAG: TRAINING-FREE RETRIEVAL-AUGMENTED ACTION GENERATION GUIDED BY GRIPPER–OBJECT GEOMETRY FROM SUCCESSFUL EXPERIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies generate actions from current observations, which can be brittle under distribution shift due to the lack of reliable guidance about future interaction outcomes. World models provide such foresight by predicting future states, but add training or test-time cost. Retrieval offers a training-free alternative, yet memory-based methods often reuse recorded actions tied to the original scene geometry, while geometry-aware alternatives typically require heavier 3D perception. We observe that successful trajectories contain two distinct signals: a scene-specific motion path and the relative gripper–object geometry that defines successful interaction. The path may become invalid after scene and position changes, while the relative geometry can remain transferable and serve as a future target even in unseen cases. Based on this insight, we propose RAAG, a training-free framework that augments frozen VLAs with future targets retrieved from successful memories. RAAG retrieves geometrically compatible experience using gripper–object relations, retargets successful relations to the current scene, and gates guidance by policy risk and retrieval reliability. The frozen VLA still determines how to reach these targets, avoiding brittle action reuse. Across five benchmarks, two simulators, and a real robot, RAAG substantially improves frozen VLAs under distribution shifts, raising OpenVLA-OFT's success rate from 24.0% to 69.6% on ManiSkill3 OOD. These gains come with modest inference overhead of 38.6% additional latency on a single GPU and only 3.0 ms on dual GPUs, where retrieval runs in parallel with VLA inference.

1 INTRODUCTION

Embodied AI aims to teach robots to robustly manipulate the physical world, widely applied to household assistance (Fu et al., 2024), warehouse logistics (Mahler et al., 2019), and assistive healthcare (Losey et al., 2022). Recent progress has been driven by vision-language-action (VLA) policies, which generate temporally extended action chunks from current observation and a language instruction (Black et al., 2024; Kim et al., 2025). While efficient, this design is brittle because actions are generated without explicitly modeling physical consequences, making small pose or appearance shifts enough to collapse performance (Fei et al., 2026; Liu et al., 2025).

World action models (WAMs) address this action-consequence link by making the future an intermediate variable for control. They predict future observations in pixel (Ye et al., 2026; Cen et al., 2025; Zhao et al., 2025; Tian et al., 2025) or latent-space (Zhang et al., 2026b; Ye et al., 2025), or learn future dynamics through large-scale pretraining (Sun et al., 2026). However, obtaining such future information requires additional learning or prediction, introducing extra expensive training or inference cost. More importantly, as shown in Fig. 1(a), errors in future prediction can directly propagate to action generation under distribution shift (Park et al., 2026; Zhao et al., 2025).

In parallel, retrieval-based methods reuse successful past experience instead of explicitly predicting the future at test time (Zhang et al., 2026c; Sridhar et al., 2025; Li et al., 2025). Yet existing methods typically retrieve visually-matched experience and reuse scene-dependent actions through replay or fusion (Kwon et al., 2025; Zhao et al., 2026). As illustrated in Fig. 1(b), visual similarity does not guarantee geometric consistency, while pose shifts can make previously successful actions non-

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

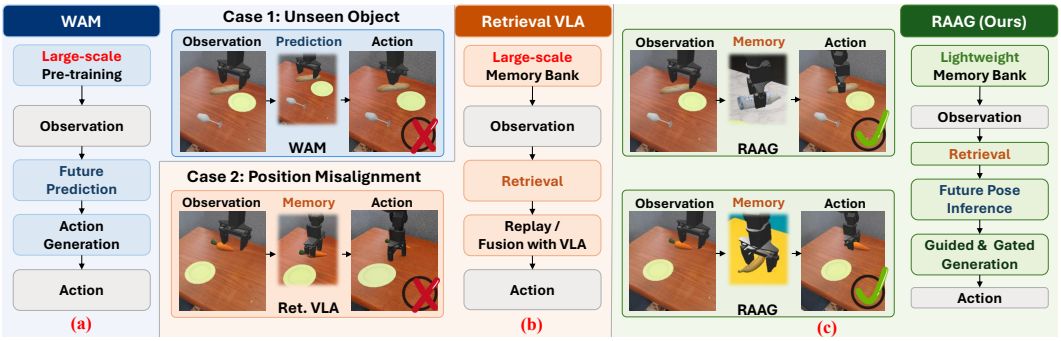


Figure 1: **Motivation of RAAG.** (a) WAMs require large-scale pretraining, while learned world predictions can become unreliable under unseen scenes or objects, as shown in Case 1. (b) Retrieval VLAs require large-scale memory banks and can fail under memory-scene misalignment in position or appearance, as shown in Case 2. (c) Our RAAG extracts transferable gripper–object relations from successful past trajectories and uses them as future guidance, enabling a lightweight memory bank to generalize experience across similarly shaped object instances and unseen scenes.

executable. Memory can therefore become unreliable precisely when the test scene differs most from past experience, and can even degrade rollouts that the base policy would otherwise solve.

The key insight is that a successful trajectory contains two distinct kinds of information, *what physical relation must be reached for success* and *how that relation was reached in a particular scene*. The latter is scene-specific, since the recorded actions depend on the original object and gripper poses and may become invalid after a scene shift. In contrast, the gripper–object relation at a successful state captures the relative direction, distance, and orientation required for interaction, and thus can remain valid after the object moves. More importantly, it comes from an actual successful execution rather than a learned prediction, making it an empirically supported future condition for the current rollout. We therefore transfer this relation rather than the recorded trajectory, and let the frozen VLA determine how to reach it in the current scene.

Building on this idea, we propose **R**etrieval-**A**ugmented **A**ction **G**eneration (RAAG), a training-free framework that augments frozen VLAs with recalled future guidance. Unlike WAMs that explicitly predict the future, and retrieval VLAs that reuse scene-dependent actions, RAAG retrieves verified gripper–object relations from past successes and uses them as future constraints for action generation. Specifically, it retrieves physically compatible experience using gripper–object geometry and object shape, then retargets successful localization and interaction relations to guide the policy toward successful interaction. The frozen VLA still generates the actions, while confidence gating controls how strongly memory intervenes based on policy risk and retrieval reliability. We extensively evaluate RAAG across five benchmarks, two simulators, and a real robot, with a particular focus on memory transferability under distribution shifts. By transferring only compact relational targets, RAAG avoids explicit future prediction and dense near-matched memory, improving success rate by up to 14.3% on standard benchmark LIBERO-Plus (Fei et al., 2026) and 45.6% on ManiSkill3 OOD (Tao et al., 2025). For discrete-token VLAs, RAAG adds 38.6% inference latency on a single GPU and only 3.0 ms on dual GPUs when retrieval runs in parallel with VLA inference.

Our contributions are summarized as follows:

1. We identify the transferable part of memory. While recorded trajectories are tied to the past scene, successful gripper–object relations can remain valid after scene changes and serve as future targets. This reframes retrieval from reusing past actions to transferring the future objective.
2. We propose RAAG, a training-free framework that realizes this idea for frozen VLAs. RAAG retrieves physically compatible memories using gripper–object geometry, retargets it to the current scene, and gates guidance by policy risk and retrieval reliability. It requires no learned future predictor and leaves action generation to the frozen policy.
3. We extensively evaluate RAAG across five benchmarks, two simulators, and a real robot under distribution shifts. RAAG improves OpenVLA-OFT's success rate from 24.0% to 69.6% on ManiSkill3, with only 3.0 ms added latency when perception model runs on an additional GPU.

2 Related Work

Retrieval and test-time guidance. Retrieval methods reuse experience through replay (Pari et al., 2021; Kwon et al., 2025), action fusion (Sridhar et al., 2025), or in-context conditioning (Zhang et al., 2026c; Li et al., 2025). However, visual similarity does not guarantee geometric compatibility. Retrieve-then-Steer (Zhao et al., 2026) uses confidence-adaptive action priors, while RA-VLA (Jang et al., 2026) learns behavior-aligned retrieval. These improve memory integration, but object-pose changes can still misalign retrieved actions. RAM (Kuang et al., 2024) addresses this mismatch through depth camera and explicit 3D affordance alignment with additional perception overhead.

Beyond retrieval, TACO (Yang et al., 2025) verifies action proposals, while ACG (Park et al., 2025) promotes temporal coherence. However, coherent actions do not guarantee geometrically valid interactions. AffordDP (Wu et al., 2025) incorporates transferable affordances into action generation, but requires training an affordance-conditioned diffusion policy. RAAG instead retrieves gripper–object targets from sparse successful trajectories to guide frozen VLAs without additional training, allowing existing motion skills to pursue interaction goals adapted to the current scene. This enables training-free transfer without closely matched memories. Further comparisons are in Appx. A.

World action models. In parallel, WAMs incorporate future information through visual prediction (Zhao et al., 2025; Cen et al., 2025; Ye et al., 2026), compact world representations (Zhang et al., 2026b), or predictive pretraining (Sun et al., 2026; Ye et al., 2025). These approaches provide learned priors for control but require large-scale predictive learning or additional future-generation computation. Moreover, visually plausible predictions need not yield executable behavior (Jiang et al., 2026). RAAG obtains future targets from the geometry of successful executions and retargets them to the current scene, providing guidance without training an additional future predictor.

3 Method

3.1 Problem Definition

We study language-conditioned robotic manipulation with a frozen VLA policy π and a fixed memory bank $\mathcal{B} = \{\tau_i\}_{i=1}^{N}$ of successful rollouts. Each trajectory records observations, robot states, and executed actions. Given an instruction, the policy predicts an action chunk from the current RGB observation and proprioceptive state. The robot executes the chunk and replans until success or budget exhaustion. Our goal is to improve task success without additional training when changes in position, appearance, or instance make the current scene differ from the stored memory.

RAAG Overview. RAAG uses successful gripper–object relations as future targets for action generation. Our motivation is that appearance changes need not alter the interaction geometry, whereas position changes can invalidate a recorded action trajectory. Transferring relative geometry therefore allows past successes to guide actions across these changes without requiring an identical memory.

As shown in Fig. 2, RAAG consists of three complementary components. *Geometry-based Retrieval (GR)* matches gripper–object relations and object shape to retrieve geometrically compatible experience, reducing sensitivity to appearance and placement. *Future-guided Generation (FG)* uses retrieved endpoint relations to specify where the gripper should reach, while the frozen VLA generates how to reach it from the current state. *Confidence Gating (CG)* adjusts guidance strength using both estimated policy risk and retrieval reliability, strengthening intervention when the policy needs support and a compatible memory is available, while limiting unnecessary or unreliable guidance.

3.2 Geometry-based Retrieval

GR aims to retrieve experience with compatible interaction geometry across changes in appearance and placement. We therefore represent each state by the object pose relative to the gripper, complemented by object shape to account for differences in contact geometry.

gripper–object relation. Interaction depends on the gripper's pose relative to the object, motivating a representation that separates interaction geometry from absolute scene placement. Let (p_t, q_t) and (p', q') denote the world-frame poses of the gripper and the target object, respectively, with positions $p_t, p' \in \mathbb{R}^3$ and orientations $q_t, q' \in \mathbb{R}^3$. Using $R(\cdot)$ to map an orientation to a rotation matrix and

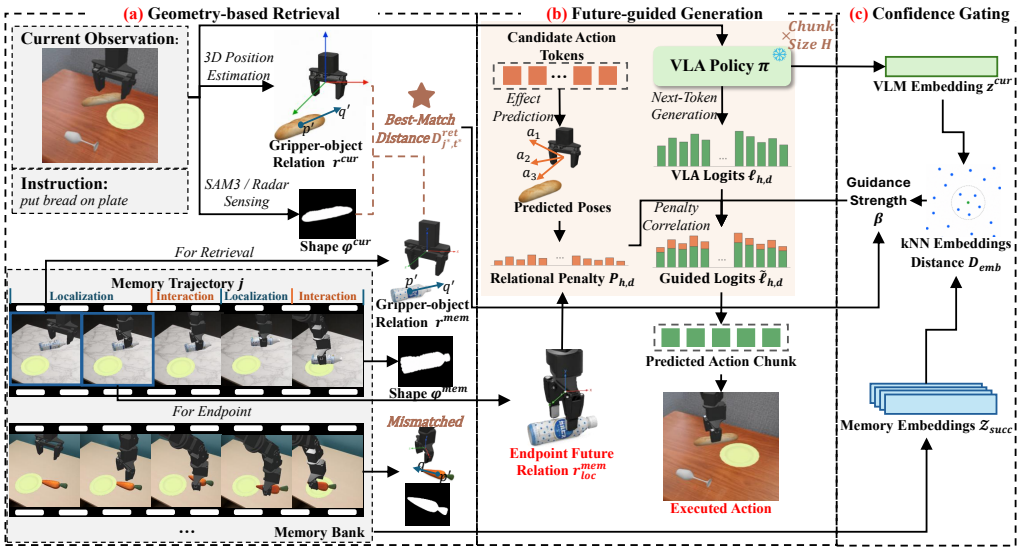


Figure 2: **Overview of our RAAG.** (a) GR retrieves the best-matched successful trajectory using gripper–object geometry and object shape. (b) FG splits execution into localization and interaction stages, uses each stage's endpoint relation as the future target, and penalizes candidate action tokens by their predicted distance to that target. Penalty computation runs in parallel with VLA inference. (c) CG combines VLA reliability and retrieval confidence to adaptively adjust the guidance strength.

$\mathrm{Log}(\cdot)$ to return its axis-angle vector, we express the object pose in the gripper frame as:

$$r = \begin{bmatrix} p^{\mathrm{rel}} \\ q^{\mathrm{rel}} \end{bmatrix}, \qquad p^{\mathrm{rel}} = R(q_t)^\top (p' - p_t), \qquad q^{\mathrm{rel}} = \mathrm{Log}\big(R(q_t)^\top R(q')\big), \tag{1}$$

where p^{rel} and q^{rel} are the relative position and orientation, respectively.

The resulting relation r supports both geometric cross-scene retrieval and endpoint transfer for action generation. To obtain it from memory and online observations, we use robot kinematics for gripper pose and a replaceable perception module for object pose. Our lightweight RGB-only implementation estimates object position and planar orientation from SAM 3 mask centroids and principal axes (Carion et al., 2026) using standard camera calibration. Details are provided in Appx. B.

Target shape. Similar relative poses can require different grasps for differently shaped objects. We therefore complement relational matching with binary silhouettes, which retain object contours without texture or background appearance. The current and memory silhouettes φ^{cur} and φ^{mem} are centered on a common grid using shared scaling that preserves aspect ratios. Their mismatch is

$$D_{\mathrm{shape}} = 1 - \frac{|\varphi^{cur} \cap \varphi^{mem}|}{|\varphi^{cur} \cup \varphi^{mem}|}. \tag{2}$$

Memory collection and standard evaluation share a fixed-camera setup, making silhouette overlap a useful, though viewpoint-dependent, cue to grasp compatibility.

Retrieval. We retrieve experience compatible with the current interaction state: relative geometry provides a prior on approach direction and interaction stage, while shape constrains grasp compatibility. Within the task-specific memory bank $\{\tau_j\}_{j=1}^{J}$, indexed by the instruction's action verb, we retrieve the trajectory and frame that minimize

$$D^{ret}_{j,t} = \left\| r^{cur} \ominus r^{mem}_{j,t} \right\|_2 + w_s D_{\mathrm{shape},j,t}, \qquad (j^\star, t^\star) = \arg\min_{j,t} D^{ret}_{j,t}, \tag{3}$$

where $\ominus$ measures the normalized geometric difference between the current relation r^{cur} and each precomputed memory relation $r^{mem}_{j,t}$ (Appx. B), and w_s weights the corresponding shape mismatch. Minimizing this score identifies a matched frame $t^\star$ in trajectory $\tau_{j^\star}$, whose subsequent endpoint provides FG's future target. For efficient retrieval, memory relations and shapes are precomputed.

Near contact, coarse relative position is insufficient because minor contact differences can change the grasp action. We therefore refine retrieval with interaction embeddings extracted by the frozen VLA

visual encoder from background-masked crops of the gripper and target. Let e^{cur} and $e^{mem}_{j,t}$ denote the embeddings of current observation and memory frame, the near-contact retrieval is operated by

$$D^{near}_{j,t} = D^{ret}_{j,t} + w_e \left\| e^{cur} - e^{mem}_{j,t} \right\|_2, \qquad (j^\star, t^\star) = \arg\min_{j,t} D^{near}_{j,t}, \tag{4}$$

where w_e weights visual evidence, complementing geometric compatibility with finer cues.

3.3 Future-guided Generation

FG uses retrieved endpoint geometry to guide motion in two stages. The localization stage brings the gripper near the object, while the interaction refines its alignment to grasp or release. Each stage uses a separate target, allowing coarse approach and precise interaction to be guided independently.

Endpoint targets. Given the best-matched trajectory retrieved by GR, FG extracts stage-specific endpoints as future targets for the current state. Gripper closing or opening marks the interaction endpoint E^{mem}_{int}, with relation r^{mem}_{int} calculated from Eq. 1. The localization endpoint E^{mem}_{loc} is the first pose along the approach satisfying $\|r^{mem}_{j,t} \ominus r^{mem}_{\text{int}}\|_2 < \delta$, using the normalized geometric residual from Eq. 3. It marks the transition from coarse approach to fine-grained interaction.

The endpoint relations $r^\star \in \{r^{mem}_{\text{loc}}, r^{mem}_{\text{int}}\}$ specifies the desired object pose at each stage's end. Because it is defined in the gripper frame, $r^\star$ transfers across changes in absolute placement.

Guided generation. FG incorporates the target gripper–object relation $r^\star$ into action generation through a geometric penalty that favors motion toward the target. Unlike directly correcting through inverse kinematics, this penalty biases the VLA's predictions while retaining its learned preferences.

For discrete Cartesian actions, FG guides next step toward the target while accounting for motion already selected within the chunk. To determine the remaining correction, we first recover the desired world-frame gripper pose from the current object pose and target relation $r^\star$. At step h, we compute the offset from the predicted pre-action gripper pose to this target. The resulting offset, expressed in the policy's action coordinates and clipped to its single-step displacement limits, defines the reference u_h. We then favor candidates that follow this correction by assigning the penalty

$$P_{h,d}(a) = \left(\frac{a - u_{h,d}}{\sigma_d} \right)^2, \tag{5}$$

where a is a candidate action value in Cartesian dimension d, and σ_d is the corresponding scale from the policy's action-normalization statistics. We incorporate this penalty into the original VLA logits $\ell_{h,d}(a)$ to guide token selection:

$$\tilde{\ell}_{h,d}(a) = \ell_{h,d}(a) - \beta P_{h,d}(a), \tag{6}$$

where CG determines the guidance strength β. After selecting all action tokens at step h, we update the predicted gripper pose and recompute u_{h+1}, keeping the next correction consistent with the selected action prefix. This design also maintains inference efficiency, as only lightweight geometric updates are performed sequentially, without additional VLA forward passes.

RAAG primarily targets discrete-token VLAs, whose logits expose candidate action probabilities for direct logits-based guidance. For other VLAs, we instead generate multiple candidate action chunks and rerank them using the same geometric guidance. This extends FG beyond token-based policies, but incurs additional inference overhead. Details are provided in Appx. C.

Stage transitions. Within the verb-specific memory bank, the retrieved trajectory's gripper-closing and opening events determine the sequence and number of localization–interaction stages. These gripper-state transitions serve as stage-completion signals and can be replaced with task-specific events for other interactions. RAAG switches from localization to interaction when no gripper pose predicted along the generated chunk further approaches the target. Executing the designated closing or opening command completes the interaction, after which RAAG advances to the next stage pair or terminates the sequence once all expected events are complete.

Interaction refinement. Near contact, small alignment errors can change the required action, demanding finer visual evidence and more frequent feedback. RAAG therefore augments geometric retrieval with local gripper–object visual embeddings (Eq. 4) to distinguish contact-sensitive configurations. To respond promptly as these configurations change, it shortens the execution horizon to H' and repeats retrieval at each action-generation round. Together, finer matching and more frequent replanning enable timely correction during grasp or release.

3.4 CONFIDENCE GATING

CG determines the guidance strength β_t in Eq. 6. Memory should intervene strongly only when the frozen VLA is outside its reliable regime and the retrieved relation is compatible with the current scene. CG therefore estimates policy confidence and retrieval confidence separately.

Policy Confidence. Motivated by prior evidence that VLA internal features distinguish successful and failing executions (Gu et al., 2026), we use the action-conditioning embedding to estimate policy risk. Taking OpenVLA-OFT (Kim et al., 2025) as an example, we obtain this embedding by mean-pooling the VLM's final hidden states before the action head, which directly condition action generation after perceiving the visual observation, instruction, and proprioceptive state.

During memory construction, RAAG stores these embeddings as $\mathcal{Z}_{succ} = \{z_i^{mem}\}$. At test time, it measures the distance from the current embedding z^{cur} to its k nearest successful anchors:

$$D_{emb} = \frac{1}{k} \sum_{z_i^{mem} \in \mathrm{kNN}_k(z^{cur}, \mathcal{Z}_{succ})} \|z^{cur} - z_i^{mem}\|_2 . \tag{7}$$

A larger distance indicates its predicted actions are more likely to fall outside the reliable behavior distribution. To obtain a bounded policy confidence, we normalize this distance as

$$c^{risk} = \frac{D_{emb}}{D_{emb} + \tau_{emb}}, \tag{8}$$

where τ_{emb} is automatically calibrated as the 95th percentile of kNN distances among successful memory states, excluding anchors from the same trajectory, and fixed within each dataset.

Retrieval Confidence. A risky policy state does not guarantee that the retrieved memory provides a transferable target. Let $D_{j^\star,t^\star}$ be the retrieval distance computed by Eq. 3 during localization and Eq. 4 during interaction. We similarly convert the distance into a bounded confidence score:

$$c^{ret} = \frac{\tau_{ret}}{D_{j^\star,t^\star} + \tau_{ret}}, \tag{9}$$

where τ_{ret} is calibrated as the 95th percentile of leave-one-trajectory-out retrieval distances among memories. It is estimated separately for localization and interaction and fixed within each dataset.

Finally, CG sets the guidance strength in Eq. 6 by scaling the maximum β_{max} with confidences:

$$\beta = \beta_{max}\, c^{risk}\, c^{ret}. \tag{10}$$

Guidance is strengthened when estimated policy risk and retrieval confidence are both high, and attenuated otherwise. When $\beta = 0$, the original action logits are preserved.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTING

Benchmarks. We evaluate RAAG across five benchmarks, two simulators, and a real robot, with a focus on its memory generalization capabilities. These include four LIBERO (Liu et al., 2023) suites, LIBERO-Plus (Fei et al., 2026) under seven perturbation factors, ManiSkill3 (Tao et al., 2025) PUTONPLATE25 with 13 OOD factors, and our FailureBench with 856 unseen failure states. We further test real-robot deployment in Appx. E and cross-environment transfer on SimplerEnv Bridge (Li et al., 2024) in Appx. I. Detailed protocols and benchmarks are provided in Appx. D.1.

Memory bank. As discussed in Sec. A, existing retrieval methods often require large banks to find near-matched trajectories rather than transferring experience across scene changes. Since prior memory banks are not released, we follow their collection protocol (Kwon et al., 2025) to collect our lightweight bank. For each suite, every training-set instruction-object-scene combination contributes one successful trajectory to the memory bank, while positions and OOD factors vary only at test time. For example, in ManiSkill3, 16 objects across 16 scenes yield 256 cases and memory trajectories. Exact memory-bank sizes for all benchmarks are reported in Appx. D.3.

Metrics. We compare against World Action Models (WAM) and retrieval VLAs. Following Li et al. (2026), we report mean episode success rate and mean latency per VLA decision.

Implementation. All experiments follow official protocols and settings of each benchmark, using one NVIDIA L40 for VLA and a second for SAM 3 (Carion et al., 2026). Reproduction training uses an NVIDIA A100. We detail the evaluated policies in Appx. D.4 and select hyperparameters by grid search on in-domain set. For GR (Eq. 3 and 4), we set w_s=0.4 and w_e=0.8. For FG, we set δ=2 and an execution horizon H'=2 for interaction stage. For CG (Eq. 7 and 10), we set K=2 and $\beta_{\max}$=10. The thresholds τ_{emb} and τ_{ret} are calibrated automatically during memory construction.

4.2 Main Results

LIBERO and LIBERO-Plus. As shown in Tab. 1, RAAG consistently improves all three frozen backbones. On LIBERO-Plus under seven perturbations, it improves OpenVLA-OFT from 66.2% to 80.5%. Even for the strong $\pi_{0.5}$, RAAG gains 13.1% and 12.5% under robot-state and layout shifts. In contrast, lighting and background shifts largely preserve interaction geometry, leaving little room for improvement. This demonstrates that recorded trajectories may no longer be executable under scene changes, whereas the geometric relation required for interaction can remain valid.

This difference is most evident against retrieval VLAs. Existing methods reuse scene-dependent trajectories, which can improve clean performance yet become harmful under distribution shift. OptimusVLA improves $\pi_{0.5}$ LIBERO performance to 98.6% but reduces LIBERO-Plus to 83.8%, whereas RAAG improves both to 99.3% and 91.3%. This is largely because RAAG using memory not to imitate, but as a transferable future goal that guides the action generation for current scene.

RAAG also differs from WAMs, where robustness depends on the reliability of the predicted future itself. This becomes a bottleneck under distribution shift, consistent with recent findings that errors in visual foresight directly propagate to action generation (Park et al., 2026). WorldVLA and DreamVLA reach only 25.0% and 48.5% on LIBERO-Plus, while VLA-JEPA reaches 79.5%. In contrast, training-free RAAG reaches 91.3% with $\pi_{0.5}$. These results suggest that effective foresight requires remaining reliable under shift, rather than simply a stronger future predictor.

ManiSkill3 and FailureBench. Tab. 2 highlights the importance of how sparse experience is transferred to a new scene. With 256 in-distribution memories, RT-Cache and VINN achieve only 2.3% and 6.2% OOD success, respectively, as replayed actions remain tied to the original scene geometry and can become misaligned after scene shifts. Our RAM adaptation addresses this limitation by re-

Table 1: **LIBERO and LIBERO-Plus success rates** (%). The upper reports standalone policies and WAMs; the lower compares base policies with add-on retrieval VLAs and our RAAG. *#Mem* denotes sum memory trajectories for all suites, and *Lat.* denotes per-decision latency (ms). Best per-backbone results are **bold**. * marks training-based methods, whose *#Mem* counts training samples.

	LIBERO					LIBERO-Plus									
Method	Sp	Ob	Go	L	**Avg**	Cam	Rob	Lang	Li	Bg	N	Lay	**All**	**#Mem**	**Lat.**
OpenVLA (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6	–	–
π_0-FAST (Pertsch et al., 2025)	96.4	96.8	88.6	60.2	85.5	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6	–	–
NORA (Hung et al., 2025)	92.2	95.4	89.4	74.6	87.9	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0	–	–
GR00T-N1.5 (Bjorck et al., 2025)	96.5	98.5	91.0	91.5	94.4	46.8	52.6	82.6	90.0	87.5	71.5	67.2	69.5	–	–
UniVLA (Bu et al., 2025)	96.5	96.8	95.6	92.0	95.2	1.8	46.2	69.6	69.0	81.0	21.2	31.9	43.9	–	–
MemoryVLA (Shi et al., 2026)	98.4	98.4	96.4	93.4	96.6	–	–	–	–	–	–	–	–	–	–
CoT-VLA (Zhao et al., 2025)	87.5	91.6	87.6	69.0	83.9	–	–	–	–	–	–	–	–	–	5953
WorldVLA (Cen et al., 2025)	87.6	96.2	83.4	60.0	81.8	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0	–	–
DreamVLA (Zhang et al., 2026b)	97.5	94.0	89.5	89.5	92.6	28.8	37.1	80.7	59.2	21.1	43.2	65.4	48.5	–	91
VLA-JEPA (Sun et al., 2026)	96.2	99.6	97.2	95.8	97.2	63.3	67.1	85.4	95.6	93.6	66.3	85.1	79.5	–	74
ForeAct (Zhang et al., 2026d)	97.3	99.8	97.3	95.4	97.5	–	–	–	–	–	–	–	–	–	–
π_0 (Black et al., 2024)	96.8	98.8	95.8	85.2	94.2	17.3	6.7	63.6	86.5	82.1	17.6	74.2	46.3	–	191.3
+ Retrieval-VLA (Zhang et al., 2026c)	97.4	98.8	96.3	89.5	95.5	–	–	–	–	–	–	–	–	800	–
+ ECHO* (Hu et al., 2026)	98.3	98.8	**98.6**	**93.5**	**97.3**	–	–	–	–	–	–	–	**56.5**	4500	–
+ **RAAG (ours, sampling-based guidance)**	**98.9**	**99.2**	97.9	93.2	**97.3**	**25.0**	**19.6**	**67.3**	**87.5**	**83.3**	**37.5**	**83.9**	55.1	**598**	413.5
$\pi_{0.5}$ (Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9	75.4	77.5	85.6	96.9	**94.6**	89.7	85.7	85.7	–	243.9
+ OptimusVLA* (Li et al., 2026)	**99.6**	**99.8**	98.4	96.4	98.6	65.7	77.7	87.1	92.2	94.5	88.5	86.6	83.8	6500	182.7
+ **RAAG (ours, sampling-based guidance)**	**99.6**	99.2	**99.8**	**98.8**	**99.3**	**79.5**	**90.6**	**87.3**	**97.3**	94.3	**94.6**	**98.2**	**91.3**	**598**	577.8
OpenVLA-OFT (L1 head; Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6	–	143.4
+ ProbeAct (Zhang et al., 2026a)	–	–	–	–	–	63.8	40.3	82.0	**93.6**	93.5	76.8	80.9	74.1	–	–
+ μVLA* (Cherepanov et al., 2026)	93.0	**99.4**	96.6	95.8	96.2	–	–	–	–	–	–	–	–	4500	150.4
OpenVLA-OFT (Discrete head)	93.3	94.1	95.9	92.4	93.9	53.2	29.2	75.4	84.9	87.8	73.0	71.7	66.2	–	147.3
+ **RAAG (ours, logit-bias guidance)**	**99.0**	99.0	**98.0**	**98.0**	**98.5**	**66.2**	**50.9**	**86.8**	90.0	**93.6**	**90.7**	**92.1**	**80.5**	**598**	151.5

Table 2: **ManiSkill3 and our FailureBench success rates** (%). ID and OOD is averaged across in-/out-of-distribution conditions. RT-Cache and VINN use VLA visual features for retrieval and then replay, while RAM directly executes transferred 3D affordances; none invokes VLA generation.

	Clean starts (official)					**From failure states** (FAILUREBENCH)						
Method	ID	Vision	Semantic	Exec.	**OOD**	ID	Position	Appear.	Adjacent	**OOD**	**#Mem**	**Lat.**
OpenVLA (Kim et al., 2024)	45.3	39.8	38.6	36.4	38.3	26.6	9.8	15.8	21.7	15.8	–	292.7
+ TT-VLA (Liu et al., 2026)	–	41.9	41.5	39.8	41.1	–	–	–	–	–	–	–
+ TraceVLA (Zheng et al., 2025)	–	59.5	45.1	26.9	43.9	–	–	–	–	–	–	–
OpenVLA-OFT (Discrete head; Kim et al., 2025)	35.9	31.2	27.6	13.3	24.0	18.8	7.1	12.2	19.9	13.1	–	84.3
+ VINN (Pari et al., 2021)	17.2	8.1	6.9	3.6	6.2	10.9	3.4	5.5	12.0	7.2	256	–
+ RT-Cache (Kwon et al., 2025)	3.1	2.8	2.5	1.6	2.3	9.4	5.5	10.1	12.2	9.1	256	31.2
+ RAM (w. our 3D perception; Kuang et al., 2024)	29.7	23.9	28.9	23.7	25.5	14.1	17.8	19.4	18.0	18.4	256	58.5
+ Robomonkey (Kwok et al., 2025)	46.9	45.2	41.2	33.1	39.8	28.1	13.4	25.4	29.9	22.6	–	319.9
+ **RAAG (ours, logit-bias guidance)**	**87.5**	**82.4**	**81.8**	**44.5**	**69.6**	**37.5**	**33.1**	**42.0**	**43.6**	**39.1**	256	87.3
OpenVLA-OFT-GRPO (Discrete head)	93.8	75.6	55.3	41.7	57.5	62.5	44.8	46.8	52.7	48.2	–	84.3
+ VINN (Pari et al., 2021)	17.2	6.9	6.6	2.1	5.2	6.2	2.8	4.2	6.7	4.5	256	–
+ RT-Cache (Kwon et al., 2025)	6.3	1.2	3.4	1.6	2.1	12.5	3.6	9.0	10.1	7.3	256	31.2
+ RAM (w. our 3D perception; Kuang et al., 2024)	20.3	24.3	30.1	23.8	26.1	12.5	15.2	21.0	18.5	17.9	256	58.5
+ Robomonkey (Kwok et al., 2025)	93.8	85.2	66.0	45.6	65.6	67.2	49.2	52.4	55.0	52.2	–	319.9
+ **RAAG (ours, logit-bias guidance)**	**96.9**	**92.8**	**87.2**	**61.5**	**80.5**	**75.0**	**59.1**	**64.9**	**67.9**	**63.8**	256	87.3
OpenVLA-OFT-PPO (Discrete head)	96.9	89.1	69.4	69.3	75.9	84.4	65.3	63.8	68.2	66.0	–	84.3
+ VINN (Pari et al., 2021)	1.6	0.0	0.9	1.0	0.7	7.8	2.2	4.5	8.5	5.2	256	–
+ RT-Cache (Kwon et al., 2025)	3.1	2.2	2.5	1.0	1.9	9.4	2.9	9.6	11.8	7.9	256	31.2
+ RAM (w. our 3D perception; Kuang et al., 2024)	18.8	26.7	29.8	24.5	27.0	14.1	20.3	22.1	21.6	21.2	256	58.5
+ Robomonkey (Kwok et al., 2025)	95.3	91.2	74.2	77.9	81.1	**89.1**	67.6	68.6	73.0	69.9	–	319.9
+ **RAAG (ours, logit-bias guidance)**	**98.4**	**95.9**	**89.1**	**84.9**	**90.0**	87.5	**76.0**	**77.7**	**77.7**	**77.0**	256	87.3

targeting the retrieved geometry to the current scene and reaching the resulting pose through visual servoing. This improves OOD success to 25.5%, but execution remains directly dependent on the accuracy of the estimated pose, making coarse perception errors consequential for control.

RAAG retains this geometric retargeting while using the retrieved endpoint as soft guidance for the frozen VLA. The geometric prior specifies the desired interaction, while the VLA's observation-conditioned action preferences help determine how to reach it. This combination raises OpenVLA-OFT's OOD success from 24.0% to 69.6% and consistently improves the stronger GRPO and PPO policies. Together, these results support transferring sparse experience through scene-adapted geometric targets, with the VLA's learned action prior reducing sensitivity to potential geometric errors.

FailureBench further increases the memory-scene mismatch by starting from unseen off-trajectory failure states, with no recovery demonstrations in memory. Here, action reuse becomes even less reliable, whereas RAAG can still use successful memories as transferable future targets, improving Base/GRPO/PPO from $13.1/48.2/66.0\%$ to $39.1/63.8/77.0\%$, respectively. This shows that geometric memory remains useful even after the object is displaced, reoriented, or even toppled.

Efficiency. RAAG obtains guidance from reusable geometric relations rather than learned future prediction, avoiding both future-generation costs and additional training. On our dual-GPU system, perception and retrieval overlap with VLA inference, leaving only a small sequential guidance overhead. The latency increases by 3.0 ms with one view ($84.3 \rightarrow 87.3$ ms) and 4.2 ms with two views ($147.3 \rightarrow 151.5$ ms). For discrete-token policies, RAAG directly biases action probabilities without additional policy forward passes, while flow-based policies require sampling and reranking, resulting in greater overhead. Appx. F provides stage-wise timings and single-GPU analysis.

Relational transfer also reduces memory requirements. The same interaction can guide actions across scene changes without densely covering each configuration. RAAG only uses 598 trajectories, compared with 800 for Retrieval-VLA, $4{,}500$ for training methods, and $6{,}500$ for OptimusVLA.

4.3 ABLATIONS

Components. According to Tab. 3(a), increasing control frequency alone provides limited improvement, while SAM 3 masking retains only the gripper and object but reduces performance to $8.8/10.1$, mainly because these masked observations differ from the policy's training inputs.

Within RAAG, replacing geometric retrieval with appearance matching lowers it to $25.5/46.2$, while retaining geometric retrieval but replacing FG with replay causes a larger drop to $16.3/21.7$. These results suggest that geometric retrieval reduces memory–scene misalignment, while geometry-

Table 3: **Ablation of RAAG.** (a) Contributions of auxiliary modifications and RAAG components. (b) Geometric compatibility in retrieval. (c) Comparisons of memory use in replaying, servoing, and guiding VLA. FB denotes FailureBench; MS3 denotes the official ManiSkill3 OOD suite.

Method	FB	MS3
Frozen policy	13.1	24.0
+ High-freq. ($H = 2$)	18.6	32.1
+ SAM 3 masking	8.8	10.1
Full RAAG	**39.1**	**69.6**
− Geometric retrieval	25.5	46.2
− Future guidance	16.3	21.7
− Confidence gating	38.0	71.0

Selection / key	FB	MS3
Fixed selection	3.8	9.5
Random selection	15.8	29.9
Appearance only	25.5	46.2
Full geometric key	**39.1**	**69.6**
− Relation	25.9	47.1
− Shape	33.8	64.0
− Appearance	38.4	68.9

Memory use	FB	MS3
Action replay w/o geo. retrieval	0.6	3.9
Action replay w. geo. retrieval	16.3	21.7
Visual servoing	18.4	25.5
Both endpoints (ours)	**39.1**	**69.6**
Localization only	31.2	53.5
Interaction only	27.5	40.7
β_{max}= 5/**10**/20 (Eq. 10)	35.1/**39.1**/37.7	58.8/**69.6**/68.9

guided generation further compensates for the residual mismatch. Finally, confidence gating controls when memory should intervene. Aggressive guidance can help weak policies but may override reliable predictions from stronger ones, motivating the competence-dependent analysis in Appx. H.

Geometry key. In Tab. 3(b), random retrieval selects one trajectory from the verb-specific bank per episode and retains it throughout, reducing success to $15.8/29.9$. Using a single fixed trajectory across all action verbs further lowers success to $3.8/9.5$. These results indicate that task compatibility alone is insufficient, the effective memory reuse also requires matching the current interaction geometry. For geometric retrieval, removing the gripper–object relation (r in Eq. 1) reduces performance to near appearance-only levels, highlighting the importance of geometric consistency for cross-scene transfer. Relative geometry alone, however, cannot account for shape-dependent grasp requirements. Shape (D_{shape} in Eq. 2) therefore constrains contact compatibility, while appearance (w_e in Eq. 4) further refines matching near contact. Further sensitivity analysis is in Appx. G.

Memory use. Tab. 3(c) compares guidance at different stages: each "only" variant applies RAAG to the named stage and leaves the other to the frozen VLA. Localization-only guidance outperforms interaction-only, suggesting that localization needs more support under scene shifts, while grasping skills transfer more readily once localized correctly. Guiding both stages further improves performance to $39.1/69.6$, demonstrating the additional benefit of refining interaction after localization.

Geometric retrieval improves action replay from $0.6/3.9$ to $16.3/21.7$, highlighting the importance of geometric consistency. Yet its limited success suggests that matching alone cannot resolve residual scene–memory misalignment. Visual servoing performs explicit 3D affordance warping equalled to RAM (Kuang et al., 2024) in Tab. 2, which is highly sensitive to quality of 3D localization. RAAG combines the geometric targets with the learned action prior, achieving $39.1/69.6$. The $\beta_{\max}$ sweep supports this balance: moderate guidance outperforms both weaker and stronger intervention.

4.4 Discussion

RAAG is limited by its lightweight RGB-only perception, which relies on SAM 3 and camera calibration. Single-view localization further assumes a tabletop, introducing height bias that multi-view triangulation can reduce (Appx. J). Future work could incorporate depth cameras, LiDARs, or mmWave radars to provide richer 3D geometry. To further validate RAAG in practical settings, we evaluate cross-environment transfer in Appx. I and real-robot performance in Appx. E. Finally, unnecessary memory intervention can disrupt strong policies. CG mitigates this risk, but more reliable policy-aware gating remains an important direction (Appx. H).

5 Conclusion

We introduce RAAG, a training-free framework that guides frozen VLAs with gripper–object relations retrieved from successful executions. By separating reusable interaction geometry from scene-dependent actions, RAAG retargets past experience to the current scene without learning a future predictor or requiring dense memories. Our results demonstrate substantially improved memory transfer across diverse appearance and pose shifts and unseen cases, highlighting the value of geometric compatibility over near-identical scene coverage. These results demonstrate the potential of a small demonstration memory to provide transferable future guidance, paving the way for embodied agents that adapt to complex new environments by reusing past successes without retraining.

AI USE STATEMENT

In this work, we used generative AI tools to assist with literature search and discovery, including identifying related work, as well as language polishing, grammar and spelling correction, and numerical consistency checks in the manuscript. We have not used generative AI tools for other research tasks requiring disclosure. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies robotic manipulation using existing benchmarks, simulated environments, and a real-robot setup and involves no human subjects or personal data. As with other robotic control methods, deployment on physical systems should include appropriate safety validation and human oversight.

REPRODUCIBILITY STATEMENT

We provide the RAAG formulation and evaluation protocol in the paper. The appendix details perception and implementation, memory construction, benchmark settings, hyperparameters, sensitivity analyses, failure cases, and cross-environment evaluation. Upon publication, we will release the code and scripts, together with the complete real-robot training dataset, trained model checkpoints, robot model files, hardware and software configuration guides, and 3D-printable models of the manipulation objects.

REFERENCES

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.

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.

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.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Egor Cherepanov, Nikita Kachaev, Daniil Zelezetsky, Aydar Bulatov, Artem Pshenitsyn, Yuri Kuratov, Alexey Skrynnik, Aleksandr I Panov, and Alexey K Kovalev. mu vla: On recurrent memory for partially observable manipulation in vla models. *arXiv preprint arXiv:2606.12497*, 2026.

Norman Di Palo and Edward Johns. Dinobot: Robot manipulation via retrieval and alignment with vision foundation models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2798–2805. IEEE, 2024.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. Safe: Multitask failure detection for vision-language-action models. *Advances in Neural Information Processing Systems*, 38:40041–40076, 2026.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

Radu Horaud and Fadi Dornaika. Hand-eye calibration. *The international journal of robotics research*, 14(3):195–210, 1995.

Yanbin Hu, Jin Cui, Jiayi Lu, Ruixuan Yang, Jun Ye, Boran Zhao, Xingyu Chen, Xuguang Lan, and Pengju Ren. Echo: Continuous hierarchical memory for vision-language-action models. *arXiv preprint arXiv:2605.10993*, 2026.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $pi_0.5$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Sanghwan Jang, Minjin Jeon, Minsoo Kim, Seongjin Choi, Dongha Kim, and Hwanjo Yu. Ra-vla: Retrieval-augmented vla for test-time adaptation. *arXiv preprint arXiv:2608.25585*, 2026.

Feng Jiang, Yang Chen, Kyle Xu, Yuchen Liu, Haifeng Wang, Zhenhao Shen, Jasper Lu, Shengze Huang, Yuanfei Wang, Chen Xie, et al. Robowm-bench: A benchmark for evaluating world models in robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4455–4460, 2026.

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

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

Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. *arXiv preprint arXiv:2407.04689*, 2024.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. Robomonkey: Scaling test-time sampling and verification for vision-language-action models. *arXiv preprint arXiv:2506.17811*, 2025.

Owen Kwon, Abraham George, Alison Bartsch, and Amir Barati Farimani. Rt-cache: Efficient robot trajectory retrieval system. *arXiv e-prints*, pp. arXiv–2505, 2025.

Runhao Li, Wenkai Guo, Zhenyu Wu, Changyuan Wang, Haoyuan Deng, Zhenyu Weng, Yap-Peng Tan, and Ziwei Wang. Map-vla: Memory-augmented prompting for vision-language-action model in robotic manipulation. *arXiv preprint arXiv:2511.09516*, 2025.

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

Zaijing Li, Bing Hu, Rui Shao, Gongwei Chen, Dongmei Jiang, Pengwei Xie, Jianye Hao, and Liqiang Nie. Global prior meets local consistency: Dual-memory augmented vision-language-action model for efficient robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35135–35145, 2026.

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

Changyu Liu, Yiyang Liu, Taowen Wang, Qiao Zhuang, James Chenhao Liang, Wenhao Yang, Renjing Xu, Qifan Wang, Dongfang Liu, and Cheng Han. On-the-fly vla adaptation via test-time reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40107–40125, 2026.

Hanqing Liu, Shouwei Ruan, Jiahuan Long, Junqi Wu, Jiacheng Hou, Huili Tang, Tingsong Jiang, Weien Zhou, and Wen Yao. Eva-vla: Evaluating vision-language-action models' robustness under real-world physical variations. *arXiv preprint arXiv:2509.18953*, 2025.

Dylan P Losey, Hong Jun Jeon, Mengxi Li, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, Jeannette Bohg, and Dorsa Sadigh. Learning latent actions to control assistive robots. *Autonomous robots*, 46(1):115–147, 2022.

Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science robotics*, 4(26): eaau4984, 2019.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. *arXiv preprint arXiv:2410.13816*, 2024.

Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

Minho Park, Kinam Kim, Junha Hyung, Hyojin Jang, Hoiyeong Jin, Jooyeol Yun, Hojoon Lee, and Jaegul Choo. Acg: Action coherence guidance for flow-based vision-language-action models. *arXiv preprint arXiv:2510.22201*, 2025.

Sangwu Park, Wonjoong Kim, Yeonjun In, Sein Kim, Hongseok Kang, and Chanyoung Park. Test-time training for visual foresight vision-language-action models. In *Continual Adaptation at Scale: Towards Sustainable AI Workshop*, 2026.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. Ricl: Adding in-context adaptability to pre-trained vision-language-action models. *arXiv preprint arXiv:2508.02062*, 2025.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, et al. Maniskill3: Gpu parallelized robot simulation and rendering for generalizable embodied ai. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 92033–92052, 2025.

Shijie Wu, Yihang Zhu, Yunao Huang, Kaizhen Zhu, Jiayuan Gu, Jingyi Yu, Ye Shi, and Jingya Wang. Afforddp: Generalizable diffusion policy with transferable affordance. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6971–6980. IEEE, 2025.

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

Yifan Yang, Zhixiang Duan, Tianshi Xie, Fuyu Cao, Pinxi Shen, Peili Song, Chenyang Zhao, Piaopiao Jin, Guokang Sun, Shaoqing Xu, et al. Fpc-vla: A vision-language-action framework with a supervisor for failure prediction and correction. *Expert Systems with Applications*, pp. 131742, 2026.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, et al. Rlinf-vla: A unified and efficient framework for vla+ rl training. *arXiv preprint arXiv:2510.06710*, 2025.

Fan Zhang, Seongbin Park, Baharan Mirzasoleiman, Shariar Talebi, and Nader Sehatbakhsh. Probe-act: Probe-guided training-free failure recovery in vision-language-action models. *arXiv preprint arXiv:2606.09740*, 2026a.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38: 24195–24228, 2026b.

Yue Zhang, Rui Wang, Jiehong Lin, Zhongrui Wang, and Xiaojuan Qi. Retrieval-vla: Training-free in-context adaptation for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1358–1367, 2026c.

Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.

Zhuoyang Zhang, Shang Yang, Qinghao Hu, Luke J Huang, James Hou, Yufei Sun, Yao Lu, and Song Han. Foreact: Steering your vla with efficient visual foresight planning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026d.

Jianchao Zhao, Huoren Yang, Hu Yusong, Yuyang Gao, Qiguan Ou, Cong Wan, SongLin Dong, Zhiheng Ma, and Yihong Gong. Retrieve-then-steer: Online success memory for test-time adaptation of generative vlas. *arXiv preprint arXiv:2605.10094*, 2026.

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*, pp. 1702–1713, 2025.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, volume 2025, pp. 54277–54296, 2025.

A Comparison with retrieval-based manipulation methods

Method	Retrieve key	Execution	Gated	RGB-only	Train-free	Memory quantity
RAM (Kuang et al., 2024)	**geometry**	3D transfer			✓	51 traj./case
DINOBot (Di Palo & Johns, 2024)	vision	replay			✓	**1 traj./case**
RT-cache (Kwon et al., 2025)	vision	replay		✓	✓	millions traj.
RiCL (Sridhar et al., 2025)	vision	prompt + fusion		✓		20 traj./task
MAP-VLA (Li et al., 2025)	vision + pose	prompt		✓		38 traj./case
OptimusVLA (Li et al., 2026)	vision + pose	fusion		✓		50 traj./case
ECHO (Hu et al., 2026)	vision + pose	fusion	✓	✓		not reported
Retrieval-VLA (Zhang et al., 2026c)	vision + pose	fusion		✓	✓	20 traj./case
AffordDP (Wu et al., 2025)	vision	**guidance**				20 traj./case
RAAG (ours)	**geometry**	**guidance**	✓	✓	✓	**1 traj./case**

Table 4: **Comparison with retrieval-based manipulation methods.** *Retrieve key* specifies the representation used to match the current state with memory. *Execution* indicates whether the retrieved experience is replayed, provided as a prompt, fused into the policy, or used as guidance. *RGB-only* denotes methods that avoid costly 3D affordance pipelines, and *Gated* marks methods that explicitly suppress unreliable memory intervention. *Memory quantity* reports the number of trajectories collected per comparable task-object-scene case, where test-time changes mainly involve object position or unseen factors.

The compared methods differ substantially in memory scale because most rely on dense coverage to find experience that closely matches the test scene. Retrieval-VLA uses 20 demonstrations per task, MAP-VLA distills approximately 378 demonstrations into trained prompts, OptimusVLA trains on 6.5K in-domain trajectories, and RT-Cache uses a million-scale bank. This scale partly compensates for the limited transferability of replayed or fused, scene-specific experience.

A small bank alone, however, does not guarantee transferable retrieval. DINOBot (Di Palo & Johns, 2024) uses one trajectory per case, but requires RGB-D perception and explicit trajectory warping. In contrast, RAAG combines a single self-collected success per comparable task–object–scene case with RGB-only perception, training-free deployment, confidence gating, and guidance-based execution. By transferring endpoint geometry rather than scene-specific actions, it enables sparse memory to remain useful under memory–scene mismatch.

B Implementation Details for Target World-Frame Position

This section describes how the target mask M_{tar} provides the position $\hat{p}_{tar}$ and orientation reference $\hat{q}_{tar}$ for relational matching and transfer used in Eq. 1. The same procedure is applied to memory observations and online queries.

Standard calibration. For the real robot, camera intrinsics K and distortion parameters are estimated using a ChArUco board with planar camera calibration (Zhang, 2000), while camera-to-robot alignment is obtained through standard hand–eye calibration (Horaud & Dornaika, 1995). To resolve the depth ambiguity of a single RGB observation, we use the tabletop as a geometric reference. Observing the calibration board placed flat on the working surface determines the plane $n^\top p + b = 0$,where $p \in \mathbb{R}^3$ is any point on the plane expressed in the robot-base frame, $n \in \mathbb{R}^3$ is its unit normal, and b determines its signed offset from the origin. Intersecting a target's viewing ray with this plane then provides its estimated world-frame position.

Single-view position. To express memory and online observations in a common geometric frame, we estimate object position using calibrated RGB geometry. Given the distortion-corrected mask centroid (u_x, u_y), camera intrinsics K, and world-to-camera extrinsics $[R_c \mid t_c]$, the viewing ray has origin $C = -R_c^\top t_c$ and direction $d = R_c^\top K^{-1}[u_x, u_y, 1]^\top$ (Hartley & Zisserman, 2003). Intersecting this ray with the calibrated table plane $n^\top p + b = 0$ gives

$$\hat{p}_{tar} = C - \frac{n^\top C + b}{n^\top d}\, d. \tag{11}$$

This estimate target object position p' in Eq. 1. Applying the same projection to memory and online observations provides a consistent reference, although elevated objects and viewpoint changes can introduce projection bias.

Normalized relational distance. Retrieval should reflect geometric compatibility without being dominated by coordinate units or rotation parameterization. We therefore define the residual used in Eq. 3 as

$$\|r_a \ominus r_b\|_2 = \sqrt{\frac{\|p_a^{\mathrm{rel}} - p_b^{\mathrm{rel}}\|_2^2}{s_p^2} + \frac{\theta_{ab}^2}{s_R^2}}, \tag{12}$$

where $\theta_{ab} = \|\operatorname{Log}((Q_a^{\mathrm{rel}})^\top Q_b^{\mathrm{rel}})\|_2$ measures angular separation and $Q^{\mathrm{rel}} = R(q_t)^\top R(q')$. For undirected planar axes, we instead use the smallest relative-angle difference modulo π. The scales s_p and s_R are calibrated as the root-mean-square positional and angular differences between valid states from distinct successful trajectories of the same task and stage, then fixed during evaluation. This balances the two terms without a manually chosen angular weight; unavailable or zero-variance terms are omitted.

Object orientation. To transfer a grasp relation across changes in object alignment, we estimate the target's planar orientation from the principal axis of its mask. Let $\bar{u}_{tar}$ be the mask centroid and v_1 the principal eigenvector of its pixel covariance. We project two points $\bar{u}_{tar} \pm \alpha v_1$, with fixed pixel span α, onto the reference plane using the same distortion correction and camera geometry as above. The resulting world-frame points $\hat{p}^{\pm}$ define

$$\hat{\theta}_{tar} = \operatorname{atan2}\big(\hat{p}_y^+ - \hat{p}_y^-,\, \hat{p}_x^+ - \hat{p}_x^-\big), \qquad \hat{q}_{tar} = [0, 0, \hat{\theta}_{tar}]^\top. \tag{13}$$

Here, $\hat{q}_{tar}$ supplies the yaw-only object-orientation reference q' in Eq. 1. The angular difference between the current and memory axes rotates the retrieved gripper orientation, preserving its alignment relative to the object rather than its absolute world-frame direction. Since the axis is undirected, angular comparisons treat orientations separated by π as equivalent.

Two-view refinement. To reduce the localization bias caused by projecting elevated objects onto the tabletop, we use a second view when both the fixed and wrist cameras observe the target. Their mask centroids define world-frame rays $C_m + \rho_m d_m$, $m \in \{1, 2\}$, where C_m is the camera center, d_m the viewing direction, and ρ_m the position parameter along the ray. We estimate the target position from the midpoint of the closest points on these rays following Hartley & Zisserman (2003):

$$\begin{aligned} (\rho_1^\star, \rho_2^\star) &= \arg\min_{\rho_1,\rho_2} \|(C_1 + \rho_1 d_1) - (C_2 + \rho_2 d_2)\|_2^2, \\ \hat{p}_{tar} &= \frac{(C_1 + \rho_1^\star d_1) + (C_2 + \rho_2^\star d_2)}{2}, \end{aligned} \tag{14}$$

where $\rho_1^\star$ and $\rho_2^\star$ locate the closest points. The wrist-camera pose is updated from the gripper pose (p_t, q_t) and the fixed hand–eye transform, requiring no recalibration as the arm moves. We retain the single-view estimate when visibility or ray geometry is unreliable, using the additional view to refine localization without making it a prerequisite. This refinement remains approximate because mask centroids need not correspond to the same physical point across views.

C Alternative Sampling-based Reranking for Future-Guided Generation

For policies without discrete action logits, FG can guide execution by selecting among complete action proposals. This alternative preserves the frozen policy's generated actions while using the retrieved target to identify a promising continuation. Guidance is applied after generation, introducing additional sampling and evaluation cost.

Geometric selection. The frozen policy generates N candidate chunks $\{a_{1:H}^n\}_{n=1}^N$ using independent noise initializations for diffusion or flow-matching heads, with the original policy output retained as one candidate. To assess their progress toward $r^\star$, we predict the gripper pose along each candidate's complete action prefixes according to its control interface. Cartesian actions update

the pose through commanded displacements, while joint actions determine configurations evaluated through forward kinematics. At each step h, the predicted pose and the fixed current object estimate yield $\hat{r}_h^n$ through Eq. 1. We jointly select the candidate and execution length that minimize the error:

$$(n^\star, h^\star) = \arg \min_{\substack{1 \le n \le N \\ 1 \le h \le H_s}} \|\hat{r}_h^n \ominus r^\star\|_2^2, \tag{15}$$

where $H_s = H$ during localization and $H_s = H'$ during interaction, and $\ominus$ is the normalized geometric residual used in retrieval.

When CG enables geometric selection, RAAG executes the prefix $a_{1:h^\star}^{n^\star}$ unchanged, preserving the coordinated actions within a single proposal. It then obtains a new observation and replans. We use $N = 3$, trading additional computation for guidance without modifying the continuous-action head.

Policy confidence for flow-based heads. To determine whether geometric selection is warranted, CG estimates policy risk from the frozen multimodal representations that condition the action expert. We obtain z^{cur} by mean-pooling the final-layer representations over valid observation and instruction tokens, excluding padding and action-suffix tokens. Because this embedding is computed before sampling, the risk estimate is independent of the noise used to generate individual proposals.

Combining this estimate with retrieval confidence as in Sec. 3.4 yields β_t. We apply geometric selection when $\beta_t/\beta_{\max} \ge 0.2$. Otherwise, we execute the original policy output with its native execution horizon, without reranking or target-based prefix truncation. Thus, the same confidence estimates that modulate logit-based guidance determine whether retrieved geometry influences proposal selection.

D Additional Implementation Details

D.1 Benchmarks

RAAG is evaluated as a test-time plug-in with all VLA weights frozen. Each backbone follows its released protocol, while the memory bank and CG thresholds are built only from successful in-distribution rollouts and fixed before evaluation.

LIBERO (Liu et al., 2023). LIBERO is one of the most widely used benchmarks for VLA evaluation. Following the standard protocol, we evaluate 40 tasks across four 10-task suites: Spatial, Object, Goal, and Long, using both third-person and wrist-camera observations. The first three test spatial, object, and goal generalization, while Long emphasizes longer-horizon manipulation. For each suite, RAAG constructs its memory bank exclusively from successful rollouts. For LIBERO-Long, we manually segment successful trajectories into ordered subtasks and store their memories separately. RAAG follows this predefined order and restricts retrieval to the current subtask, within which geometric matching determines the object to manipulate and the placement target. Repeated retrieval remains within the same subtask until execution advances to the next.

LIBERO-Plus (Fei et al., 2026). LIBERO-Plus extends LIBERO into a large-scale robustness benchmark with 10,030 perturbed task instances across seven factors: camera, robot state, language, lighting, background, sensor noise, and layout. We evaluate all factors using the **same** LIBERO memory bank, without adding any perturbed rollouts. For camera-view perturbations, we obtain the updated camera calibration from the simulator for geometric estimation. This directly tests whether fixed in-distribution experience transfers under broad visual, geometric, and semantic shifts.

SimplerEnv Bridge (Li et al., 2024). SimplerEnv Bridge tests portability to a new simulator and robot. We evaluate four visual-matching tasks using a single third-person camera view, with 3 runs and 24 episodes each, totaling 288 episodes per backbone. Memory contains only successful executions from the official configurations and remains fixed during evaluation.

ManiSkill3 (Tao et al., 2025). We use PutOnPlate25, a 25-task pick-and-place benchmark with diverse objects and receptacles, using a single third-person camera view. The adopted protocol contains one in-distribution setting and 13 OOD conditions covering appearance changes, unseen objects, receptacles and instructions, and altered object or robot states. One bank per model, collected from that checkpoint's own successful ID rollouts and fixed across all OOD conditions.

Our FailureBench. Recovering from genuine VLA failures is substantially harder than acting from clean initial states. The policy must continue from a state produced by an unsuccessful rollout, while no failed trajectories or recovery demonstrations are available. We therefore introduce FAILUREBENCH, in which each policy starts from the exact simulator state where a frozen VLA previously failed. Moreover, the memory bank contains only successful trajectories, directly testing whether successful experience can transfer to unseen failure states rather than simply replaying near-matched recovery behavior.

Built from OpenVLA-OFT (Kim et al., 2025) failures across the 14 ManiSkill3 PUTONPLATE25 suites, FAILUREBENCH is constructed by running OFT-base with an 80-step budget and retaining the terminal states of unsuccessful episodes. We exclude physically unrecoverable cases, such as objects knocked off the table, resulting in 856 recoverable states. As illustrated in Fig. 3, simulator ground truth groups these states into three failure types: *appearance change*, where the target is toppled or flipped; *adjacent object*, where it touches or closely neighbors another object; and *unusual position*, where it lies far from its typical training location or presents a difficult grasp pose. We evaluate 856 states, comprising 207, 328, and 321 cases of these three types, respectively.



Figure 3: **Examples of our FailureBench.** (a) Appearance change. (b) Adjacent object. (c) Unusual position.

D.2 RUN-TO-RUN VARIATION

We repeat three times to estimate evaluation noise. The observed mean absolute deviation is 0.9% on FailureBench, 0.4% on the ManiSkill3, 0.4% on LIBERO, and 0.3% on LIBERO-Plus.

D.3 MEMORY BANK

To make the memory budget explicit, Tab. 5 reports the number of trajectories used for each benchmark, which is substantially smaller than in prior retrieval-based methods. Each trajectory records one successful rollout, including its task instruction, RGB observations, and executed actions at each step. RAAG further precomputes the target mask and gripper–object relation at each frame and stores the action-conditioning embeddings used by CG to improve the test-time efficiency.

Table 5: **Number of memory trajectories used for each benchmark.** LIBERO-Plus reuses the memory bank of LIBERO.

Benchmark	ManiSkill3	FailureBench	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	SimplerEnv Bridge
Memory size	256	256	170	152	40	236	37

D.4 POLICIES

We evaluate RAAG with three frozen policy backbones: OpenVLA-OFT (Kim et al., 2025), π_0 (Black et al., 2024), and $\pi_{0.5}$ (Intelligence et al., 2025). The flow-based π policies use the sampling-based FG variant described in Appx. C.

For official OpenVLA-OFT, we evaluate both a discrete action head and a continuous L1-regression head. The L1 head achieves higher performance at comparable inference cost (Kim et al., 2025), but does not expose action probabilities for direct logit-bias guidance. Applying RAAG to this head would therefore require the more computationally expensive sampling-based variant. We instead apply RAAG to the discrete head, which supports efficient probability-level guidance, and report unguided results for both heads to separate the benefit of guidance from the action representation.

On ManiSkill3, we additionally evaluate the GRPO (Shao et al., 2024), and PPO (Schulman et al., 2017) OpenVLA-OFT checkpoints released by RLinf (Zang et al., 2025), spanning different levels of policy competence. We construct separate memory banks from each model's successful rollouts.

E Real-Robot Evaluation

We evaluate whether RAAG can improve a frozen policy's real-world pick-and-place performance under varied object placements and perturbations. To isolate the effect of retrieval guidance, we use the same demonstration collection and policy checkpoint for the comparisons with and without RAAG.

Robot setup. We use the open-source LeRobot SO-ARM101, with STS3250 servos replacing the shoulder-lift and elbow-flex servos to increase payload capacity, while the remaining joints retain standard STS3215 servos. A fixed overhead RGB camera provides workspace context, while a wrist-mounted RGB camera provides a close view of the grasp region. To connect visual observations to geometry, we calibrate the camera intrinsics and camera-to-robot transforms using a ChArUco board, together with the table plane and encoder-to-URDF joint offsets. This establishes the common coordinate system needed to compare remembered gripper–object relations with poses predicted by forward kinematics.

Objects and demonstrations. We collect 40 successful demonstrations for each of three open-source 3D-printed objects—a carrot, a banana, and an orange—yielding 120 trajectories recorded at 10 Hz through leader–follower teleoperation. The spherical orange is particularly sensitive to grasp alignment, as it can easily slip out as the gripper closes, demanding precise localization and positioning. Initial placements vary, and each scene contains an instructed target alongside two distractors as shown in Fig. 4(a). The instruction, *Put the [object] on the black plate*, therefore requires selecting a specific object rather than simply grasping an available one. We use those demonstrations to both fine-tune the base policy and construct RAAG's memory bank, giving RAAG no additional demonstration data. This allows us to test whether retrieving information from demonstrations already used for training can further improve the policy's robustness across different test conditions when demonstrations are limited.

Policy training. We initialize from OpenVLA-7B (Kim et al., 2024) and adapt the official OpenVLA-OFT (Kim et al., 2025) training implementation to the SO-ARM101. The policy takes the instruction, two 256×256 RGB observations, and the robot's proprioceptive state. Feature-wise Linear Modulation (FiLM; Perez et al., 2018) conditions visual features on the instruction to support target selection in multi-object scenes.

We use the parallel discrete-action variant, which exposes action probabilities for RAAG's decoding guidance. Each query predicts a 10-step chunk of Cartesian end-effector actions and gripper commands, with 256 bins per action dimension. Demonstration actions are converted to end-effector space using forward kinematics. At deployment, RAAG guides the Cartesian action predictions, which are converted into joint commands through LeRobot's URDF-based inverse-kinematics interface. We fine-tune with action-token cross-entropy, rank-32 LoRA, and AdamW for 20 epochs, using a learning rate of 5×10^{-4} and an effective batch size of 64 following the standard practice (Kim et al., 2025). Both evaluation methods use the same frozen checkpoint and inverse-kinematics controller.

RAAG configuration. RAAG uses SAM 3 (Carion et al., 2026) to obtain object and plate masks from the overhead observation. Following the detect-then-track strategy of Grounded SAM (Ren et al., 2024), we perform open-vocabulary segmentation on the first frame and propagate the masks through subsequent frames, maintaining temporal correspondence without repeating open-vocabulary detection.

To use these observations for geometric retrieval, we back-project the masks onto the calibrated table plane to estimate object and plate coordinates in the robot base frame, and obtain the gripper pose through forward kinematics. Expressing demonstrations and test observations in this common frame allows RAAG to compare their gripper–object relations and retrieve a compatible successful configuration. The retrieved relation is then instantiated at the object's current location to guide the policy's action predictions. In this way, a demonstration collected at one placement can provide useful guidance at another, without replaying its original joint trajectory.

Table 6: Real-robot pick-and-place results on seen black plates and unseen white plates. Both methods use the same discrete OpenVLA-OFT checkpoint trained exclusively on black-plate demonstrations. We report success rate (%), with overall results pooled across the three objects. Completion time is measured on successful trials, while decision latency is measured across all policy queries.

Target plate	Method	Carrot	Banana	Orange	Overall	Completion time (s) ↓	Decision latency (ms) ↓
Black (seen)	OpenVLA-OFT	45	40	15	33.3	19.3	174.2
	OpenVLA-OFT + RAAG	80	70	35	61.7	15.8	177.6
White (unseen)	OpenVLA-OFT	20	10	0	10.0	22.5	174.2
	OpenVLA-OFT + RAAG	75	70	25	56.7	20.6	177.6

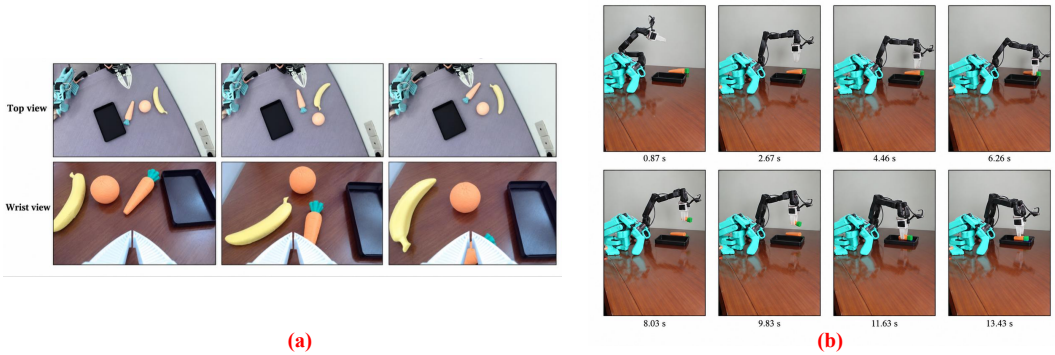


Figure 4: Real-robot demonstrations and task execution. **(a)** Three training examples with different object arrangements, shown from the overhead and wrist cameras. In each example, one object is the instructed grasp target and the remaining objects serve as distractors. **(b)** An example test-time pick-and-place sequence with execution progression timestamps.

Evaluation protocol. We evaluate whether RAAG improves both task execution under familiar conditions and generalization to an unseen placement target. All demonstrations use a black plate, whereas evaluation includes both black and white plates. The white-plate condition changes the target's appearance while preserving the pick-and-place objective, allowing us to examine whether guidance based on geometric relations remains useful beyond the demonstrated target appearance. The instruction specifies the plate color in each condition.

For each object and plate, we evaluate 20 distinct initial positions with and without RAAG. We alternate the two methods and restore the initial scene as closely as possible between paired attempts to reduce the influence of scene variation and temporal drift. Both methods share the same observation processing and robot execution settings. Success requires the robot to grasp the instructed object and release it onto the specified plate, where it remains supported without human assistance.

Results. Table 6 shows that RAAG improves success under both plate conditions, with the largest gain on the unseen white plate: from 10.0% to 56.7%. The substantial degradation of the direct policy under this unseen target condition suggests sensitivity to visual features specific to the training setting. By providing a geometric prior retrieved from demonstrations, RAAG reduces this sensitivity and supports more robust transfer to the unseen plate.

RAAG also shortens completion time on successful trials with only 3.4 ms of additional decision latency. This is consistent with more accurate localization reducing corrective motions and repeated grasp attempts, although the successful trial subsets differ between methods.

F Inference Cost

RAAG's efficiency comes from overlapping memory access with policy inference rather than adding a sequential future-generation stage. As shown in Table 7, the retrieval pipeline takes 52.6 ms, primarily spent on object segmentation with the SAM 3 (Carion et al., 2026) perception foundation

Table 7: **Per-decision latency breakdown** in milliseconds. ‖ denotes stages executed in parallel with the VLA forward pass, so Total follows the critical path. WM data denotes the samples used to train a world model or failure supervisor. "Generation" records the additional latency incurred outside the VLA generation. Parentheses denote serialized/single-GPU latency.

Method	WM data	VLA	Retrieve	Generation	Gating	Total
OpenVLA-OFT (frozen) (Kim et al., 2025)	0	147.3	—	—	—	147.3
VLA-JEPA (Sun et al., 2026)	296K	74	—	—	—	74
CoT-VLA (Zhao et al., 2025)	311K	108	—	5845	—	5953
DreamVLA (Zhang et al., 2026b)	76K	88	—	3	—	91
FPC-VLA (Yang et al., 2026)	110K	180	—	1590	—	1770
μVLA (Cherepanov et al., 2026)	0	143.4	—	6.0	—	150.4
OptimusVLA (Li et al., 2026)	0	179.2	—	3.5	—	182.7
RAAG (ours)	0	147.3	$52.6^{\|}$	1.3	2.9	$151.5^{\|}$ (204.1)

Table 8: **Sensitivity to memory budget and retrieval parameters** on ManiSkill3.

	FailureBench	ManiSkill3
(A) Memory budget		
None	13.1	24.0
1 case	3.8	9.5
16 cases	20.1	47.8
32 cases	27.5	56.0
64 cases	31.4	62.9
128 cases	34.3	66.3
256 cases	**39.1**	**69.6**

	FailureBench	ManiSkill3
(B) Retrieval parameters		
$\delta = 1$	31.4	69.2
$\delta = \mathbf{2}$	**39.1**	**69.6**
$\delta = 4$	23.9	56.5
$w_s = 0$	33.8	64.0
$w_s = \mathbf{0.4}$	**39.1**	**69.6**
$w_s = 0.8$	33.9	69.2

model, and runs concurrently with the 147.3 ms OpenVLA-OFT forward pass on our dual-GPU system. Only guidance and validation remain sequential, adding 4.2 ms in total and yielding a decision latency of 151.5 ms—a 2.9% increase over the base policy. RAAG thus supplies an experience-based geometric prior without training a world model or explicitly generating future observations.

In real-world deployment, SAM 3 perception runs on the retrieval branch. Its cost can likewise be hidden when this branch completes before policy inference. This small critical-path overhead therefore depends on concurrent execution: on a single GPU, resource contention or serialized execution can expose additional perception and retrieval costs.

G Retrieval Stability and Sensitivity

In addition to Sec. 4, we further examine whether RAAG retrieves stable memories using different memory banks and hyperparameters in Tab. 8.

Memory budget. RAAG benefits from a small but sufficiently diverse memory bank. With only one case, limited interaction coverage can produce mismatched targets, while CG lacks the cross-trajectory samples needed for reliable calibration. Unreliable guidance can therefore override useful policy predictions, consistent with success dropping from $13.1/24.0$ to $3.8/9.5$. Expanding the bank to just 16 cases raises success to $20.1/47.8$, capturing $27\%/52\%$ of the full-bank improvement. Larger banks generally provide further gains, reaching $39.1/69.6$ with 256 cases. These results suggest that RAAG needs sufficient coverage for compatible retrieval and confidence calibration, but can transfer successful relations without densely storing near-matched trajectories.

Endpoint precision and object shape. A tight endpoint is essential. Relaxing δ from 2 to 4 reduces success from $39.1/69.6$ to $23.9/56.5$, while an overly strict 1 threshold also hurts FailureBench. Removing shape matching reduces performance by 5.3 and 5.6 points, whereas increasing its weight to 0.8 provides no further benefit. These results show that geometry is the primary retrieval signal, while object shape serves as a complementary cue for selecting an interaction-compatible relation. The deployed values were fixed using in-distribution memory and reused across all benchmarks without per-benchmark tuning.

Table 9: **Failure-prediction AUROC of the two CG signals.**

Setting	Policy	D_{emb}	D_{rel}
ManiSkill3	Base	**0.71**	0.52
	GRPO	**0.89**	0.46
	PPO	**0.88**	0.83
FailureBench	Base	**0.76**	0.48
	GRPO	**0.80**	0.47
	PPO	**0.82**	0.70

Table 10: **Sensitivity to the nearest neighbors for confidence gating.**

k	Base	GRPO	PPO	Avg.
0	**71.0**	74.3	81.9	75.7
1	67.6	76.8	**91.4**	78.6
2	69.6	**80.5**	90.0	**80.0**
3	66.7	74.1	83.3	74.7
4	63.3	70.8	76.5	70.2
5	61.0	67.1	59.7	62.6

H Mechanism Analysis on Confidence Gating

RAAG should intervene only when the frozen policy is unreliable. Thus, CG must judge whether an action proposal is likely to fail without training a separate detector. We evaluate two signals by replacing z^{cur} and $\mathcal{Z}_{succ}$ in Eq. 7. The embedding signal uses the frozen VLA representation that jointly encodes the observation, robot state, and instruction and directly conditions action generation. A large distance from successful embeddings therefore indicates an unfamiliar and potentially unreliable proposal. The relational signal instead compares the gripper-target relation induced by the proposal with successful interaction relations, directly detecting geometrically abnormal actions.

Tab. 9 shows that embedding distance is the stronger general risk signal, reaching AUROC $0.89/0.88$ on fresh OOD rollouts and $0.80/0.82$ on genuine failure states for GRPO/PPO. Relational distance is weaker overall because it captures only interaction geometry, which varies with the task and initial robot state, whereas embedding distance jointly accounts for these factors. We therefore use embedding distance to estimate general policy risk and retrieval distance to detect geometrically unreliable proposals.

In Tab. 10, we further vary the number of nearest successful embeddings K (used to compute D_{emb} in Eq. 7) for policies with different competence, where $K{=}0$ disables CG. For the weak Base policy, memory guidance is often beneficial, so gating can unnecessarily suppress reliable retrieval. In contrast, stronger GRPO and PPO policies benefit from CG by avoiding unnecessary memory intervention, while a large K introduces less relevant success states and weakens the local risk estimate. Overall, a small neighborhood ($K{=}2$) provides the best balance across policy strengths.

I Cross-Environment Evaluation on SimplerEnv

SimplerEnv evaluates whether RAAG transfers beyond its original simulator. Because OpenVLA-7B achieves limited success in this setting, we construct its retrieval memory from successful VLA-JEPA rollouts. As shown in Tab. 11, RAAG improves all three backbones, raising OpenVLA-7B from 1.0% to 24.3%, SpatialVLA from 42.7% to 50.0%, and VLA-JEPA from 57.3% to 66.0%. This result directly supports our central claim that the reusable part of successful experience is the interaction relation rather than the policy-specific action sequence. RAAG also improves OpenVLA substantially more than V-GPS, another zero-shot plug-in method reported under the same protocol, reaching 24.3% versus 6.5%. These results show that relational guidance transfers across policy architectures and action-generation mechanisms without retraining.

J Failure Analysis

RAAG obtains object geometry directly from RGB observations without a trained depth model. With a single camera, however, a mask centroid defines only a viewing ray. We therefore intersect this ray with the table plane to estimate the object position. This is accurate when the object lies on the table, but becomes biased when the visible centroid is above the assumed plane, as shown in Fig. 5(c). A second camera removes this assumption by triangulating the object from two viewing rays.

Table 11: **Cross-policy transfer on SimplerEnv Bridge.** Success rate (%) averaged over three runs, each with 24 episodes per task, across four visual-matching tasks.

Model	Spoon	Carrot	Stack	Eggplant	Avg.
Frozen zero-shot policies					
OpenVLA-7B (Kim et al., 2024)	0.0	0.0	0.0	4.1	1.0
OpenVLA-7B + V-GPS (Nakamoto et al., 2024)	0.0	4.0	2.0	**20.0**	6.5
OpenVLA-7B + RAAG (ours)	**41.7**	**37.5**	**8.3**	9.7	**24.3**
SpatialVLA (Qu et al., 2025)	16.7	25.0	29.2	**100.0**	42.7
SpatialVLA + RAAG (ours)	**33.3**	**33.3**	**33.3**	**100.0**	**50.0**
VLA-JEPA (Sun et al., 2026)	**75.0**	70.8	12.5	70.8	57.3
VLA-JEPA + RAAG (ours)	70.8	**77.8**	**33.3**	**81.9**	**66.0**

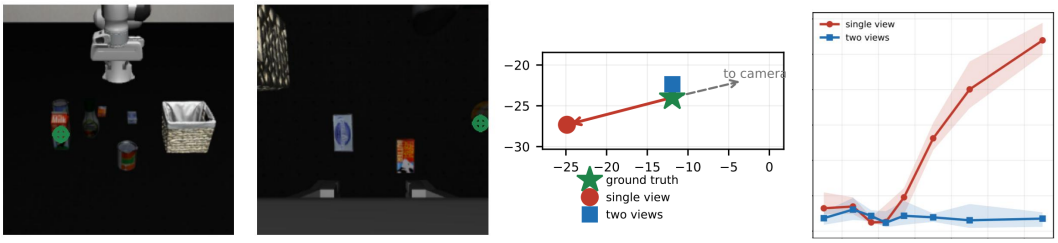


Figure 5: **Single-view depth bias and two-view triangulation.** (a) A single workspace camera provides only a viewing ray through the target mask centroid. (b) A wrist camera supplies a second ray, allowing the target position to be triangulated. (c) An example for top-down localization in world coordinates, where the horizontal and vertical axes show the object's world x and y positions in centimetres. The markers compare the ground-truth position with the single-view and two-view estimates. (d) Localization error over 5,740 frames. The horizontal axis is the object height bins, and the vertical axis is the resulting 3D localization error in centimetres. The curves show the median error, and the shaded regions indicate the interquartile range.

Table 12: **Object localization error under different object heights.** Median error in centimetres for single-view plane intersection and two-view triangulation on LIBERO frames.

Object height above the assumed plane	Frames	Single view	Two views	Reduction
resting on the table (< 2 cm)	4,161	1.19	**0.30**	75%
2–5 cm	745	2.78	**0.47**	83%
5–10 cm	285	11.12	**0.25**	98%
> 10 cm	549	22.42	**0.18**	99%
all frames	5,740	2.25	**0.29**	87%

We compare single-view plane intersection and two-view triangulation on the same 5,740 frames from all four LIBERO suites. Both methods use the same segmentation centroid and are evaluated against the simulator ground truth, so their difference directly measures the cost of the single-view depth assumption. As shown in Tab. 12, the single-view estimate is reasonable when the object is close to the table, with 1.19 cm error below 2 cm height. Its error then grows rapidly to 11.12 cm at 5–10 cm and 22.42 cm above 10 cm. In contrast, two-view triangulation remains near 0.30 cm across all height ranges. Overall, it reduces the median error from 2.25 to 0.29 cm.

Fig. 5(d) further shows that the single-view error grows with object height, isolating the main geometric limitation of RAAG. The transferable relation remains useful, but its retargeting becomes inaccurate when object depth is estimated from an invalid scene assumption. Multi-view triangulation resolves this failure without learned depth prediction or additional policy training. However, triangulation requires the object to be visible from both cameras, so single-view plane intersection is retained as a fallback.