GENIEMASTER: AN EMBODIED CONTINUAL HARNESS FOR PERSISTENT MULTI-ROBOT WORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Persistent robot work requires general reasoning, responsive manipulation, and reliable recovery as tasks and environments change. Frontier agents provide high-level reasoning, while learned policies provide continuous, closed-loop control from current observations. We present GENIEMASTER, an embodied continual harness for persistent multi-robot work. At its core is the Embodied Execution Protocol (EEP), an interface through which the agent interacts with the physical world. Through EEP, the agent invokes robot capabilities and guides policies with spatial cues, object references, and detailed language. We co-design the guidance interface and policy training so that policies can translate agent-provided guidance into robot actions. The agent uses physical feedback as evidence for task progression, guidance revision, and recovery, while retained task state supports coordination across robots and resumption after interruptions. GENIEMASTER supports continual harness evolution through asynchronous reflection and memory consolidation, using execution experience to update guidance, assessment criteria, and orchestration strategies. Across six real-world tasks, GENIEMASTER achieves 78.67% macro-averaged success versus 38.83% for the Dual-system baseline based on a vision–language model. Further evaluations show improved retrieval of unseen products with object-reference conditioning, while a convenience-store deployment illustrates ongoing task orchestration over a 24-hour window as new requests arrive.

1 INTRODUCTION

How can the generalization of frontier agents translate into dexterous, responsive robot behavior? An interface must allow agent-acquired context to shape the actions of policies and controllers, and make physical outcomes available as evidence for subsequent reasoning. These exchanges are embedded in ongoing work: each decision depends on prior progress, dependencies across robots, and accumulated experience. An embodied agent harness organizes this interaction over time, connecting individual capability invocations to the continuity of work across tasks and robots.

We present GENIEMASTER, an **embodied continual harness** for persistent multi-robot work. At its core is the **Embodied Execution Protocol (EEP)**, an extensible, bidirectional interface through which a coding agent interacts with robot capabilities and their execution outcomes. Building on hierarchical execution (Liang et al., 2022; Shi et al., 2025), the agent invokes policies and control tools, supplies guidance, and uses physical feedback to manage progression and recovery. Retained assignments, progress, and dependencies support coordination across robots and resumption after interruptions.

To make guidance supplied through EEP usable by learned policies, we align policy training with the conditions the agent provides during execution. Demonstrations pair spatial cues, object references, and detailed instructions with the corresponding actions, teaching policies to use these conditions alongside current observations. This design builds on language- and spatially conditioned control (Belkhale et al., 2024; Dai et al., 2024; Li et al., 2025) and allows the agent to guide execution with context obtained from observations or local and online assets. The policy remains responsive between agent updates, while success/error detection provides evidence for revising guidance and recovering from failures.

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.

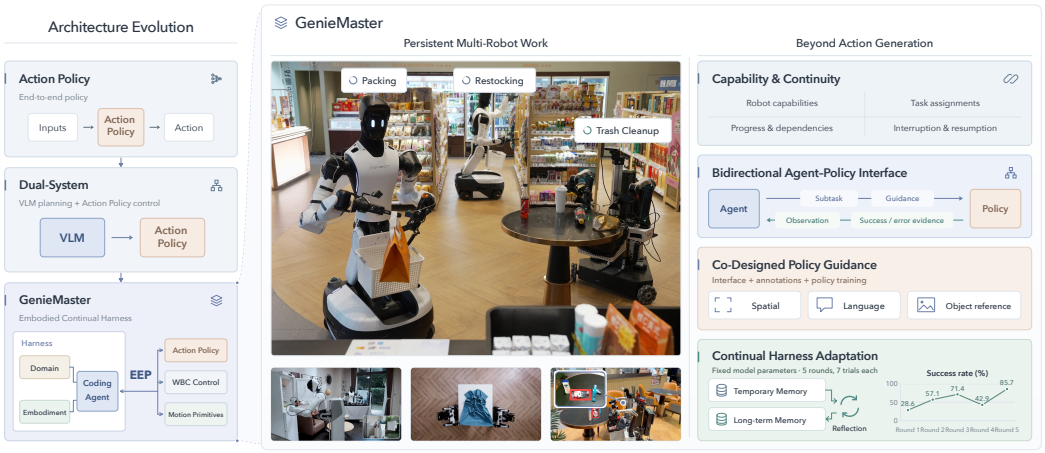


Figure 1: **GenieMaster overview. Left:** From VLA and dual-system control to an embodied continual harness. The Embodied Execution Protocol (EEP) serves as an agent–embodiment interface. **Center:** Persistent multi-robot work. **Right:** Capability invocation and task continuity, the bidirectional agent–policy interface, co-designed guidance, and experience-based adaptation with model parameters fixed. The curve reports office-supply search success over five rounds.

Asynchronous reflection turns execution records into reusable guidance, assessment criteria, and recovery strategies (Zhang et al., 2026b; Karten et al., 2026; Ding et al., 2026). Temporary memory stores recent records, while long-term memory retains consolidated lessons for subsequent guidance, completion assessment, and orchestration. Reflection runs separately from online detection, with model parameters fixed during deployment.

Across six real-world tasks, GENIEMASTER achieves 78.67% macro-averaged success versus 38.83% for the Dual-system baseline based on a vision–language model (VLM). With policy fine-tuning on 1,000 stock-keeping units (SKUs), EEP couples agent-supplied guidance with policies trained to use it, raising zero-shot retrieval success on held-out products from 20% for the text-only baseline to 92%. On LIBERO-Pro, the EEP-based GENIEMASTER workflow raises success from 10.75% for the base policy to 60.38%. EEP thereby supports in-context learning, allowing guidance-trained policies to use agent-supplied knowledge and instructions at execution time rather than relying solely on information encoded in their parameters.

Our contributions are:

- **An embodied agent harness for persistent multi-robot work.** We introduce GENIEMASTER, centered on the Embodied Execution Protocol (EEP), an extensible, bidirectional interface between the agent and the physical world. Physical feedback supports task progression and recovery, while retained task state enables multi-robot coordination and resumption after interruptions.

- **Agent–policy co-design for actionable multimodal guidance.** We train policies on demonstrations annotated with the same forms of spatial cues, object references, and detailed language that the agent supplies during execution. This alignment enables policies to use agent-acquired context alongside current observations, including object references for products unseen in policy fine-tuning. A success/error detector returns evidence of physical outcomes, which the agent uses to revise guidance and recover execution.

- **Continual harness evolution through execution experience.** Asynchronous reflection and memory adapt subsequent guidance, completion assessment, and orchestration while model parameters remain fixed during deployment.

2 Related Work

Interfaces between reasoning and robot control. Programmatic interfaces let language agents compose perception and control primitives (Liang et al., 2022; Fu et al., 2026). Constraint-based approaches express task intent as geometric objectives for motion planning (Huang et al., 2024; Pan et al., 2025; Zhang et al., 2026a). Learned policies provide continuous observation–action control (Brohan et al., 2023; Kim et al., 2024; Physical Intelligence et al., 2025), with hierarchical systems linking reasoning to execution through skill selection and instructions (Ahn et al., 2022; Shi et al., 2025). Beyond subtask instructions, richer policy interfaces convey detailed language (Belkhale et al., 2024; Dai et al., 2024), spatial paths (Li et al., 2025), point or box prompts (Wang et al., 2026c), and subgoal images (Physical Intelligence et al., 2026). Building on these interfaces, GenieMaster co-designs agent-provided guidance with demonstration annotations and policy training, so that policies can use object references and spatial cues on historical frames alongside current observations. EEP returns physical feedback to inform subsequent guidance and execution decisions.

Embodied-agent harnesses and experience-driven improvement. Feedback-based planning and skill-handoff studies use execution outcomes to guide progression and recovery (Huang et al., 2022; Hu et al., 2026; Lei et al., 2026; Rui et al., 2026). Embodied-agent harnesses organize monitoring, recovery, and memory of capability boundaries (Wang et al., 2026b; Yoon et al., 2026; Zhang et al., 2026b; Huang et al., 2026). Experience can adapt skills, configurations, and runtime critics (Wang et al., 2023; Karten et al., 2026; Ding et al., 2026), with recent work studying retention-aware and evidence-gated updates (Kang et al., 2026; Wang et al., 2026a). GenieMaster connects these execution-management mechanisms to policy guidance through EEP: physical outcomes inform immediate guidance revision and task progression, while reflection consolidates execution experience into instructions for later tasks.

Multi-robot coordination and persistent operation. Language-model task allocation (Kannan et al., 2023), shared spatial knowledge (James et al., 2026), and memory-supported assistance (Jin et al., 2026) complement long-term service autonomy (Hawes et al., 2016). GenieMaster connects these coordination needs to learned manipulation: physical outcomes update progress and dependencies, and retained task state supports resumption across invocations and robots. This continuity concerns unfinished work, while experience memory informs how subsequent work is performed.

3 GenieMaster

Figure 2 shows how GenieMaster organizes robot execution. We first describe EEP's contracts, then explain how training makes agent guidance usable by policies and how reflection and memory inform later decisions.

3.1 Embodied Execution Protocol

EEP defines how the agent interacts with the physical world through robot capabilities. It organizes these interactions through four contracts: **Capability** exposes operations and their invocation interfaces; **Guidance** specifies policy instructions and conditions; **Feedback** returns evidence of physical outcomes; and **Continuity** specifies the task state retained across invocations.

Invoking capabilities. Following the Model Context Protocol (MCP) tool abstraction (Model Context Protocol Contributors, 2025), the **Capability contract** exposes tool descriptions, inputs, and results for selection in the current task context. **Domain** tools compose workflows, while **Embodiment** implementations bind capabilities to robot-specific policies and controllers. Item retrieval, for example, composes `navigation`, `pick`, and `place`; individual tools remain available for

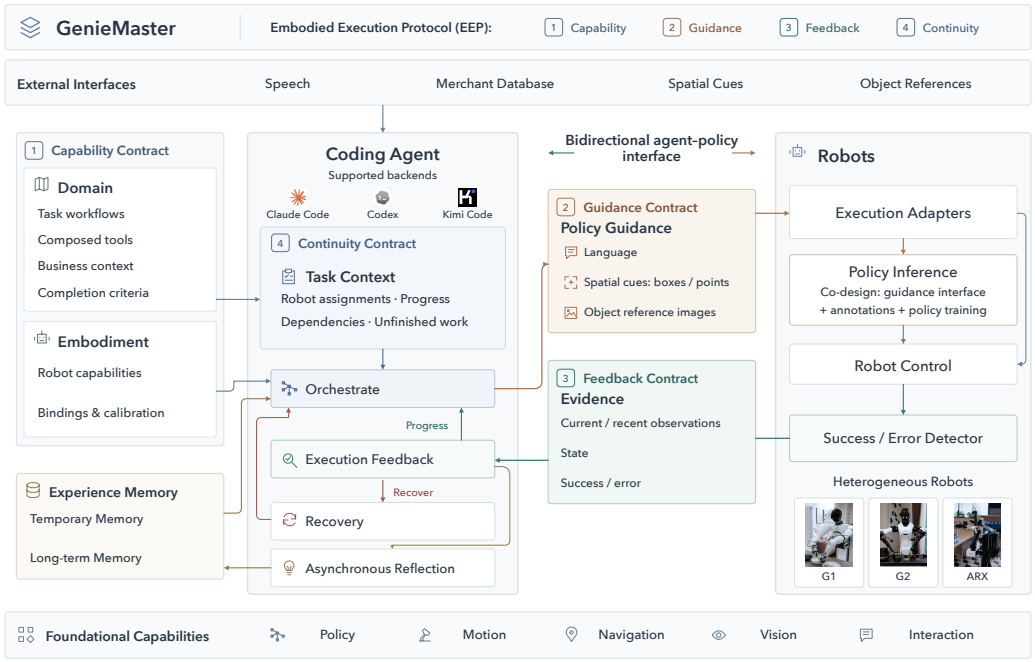


Figure 2: **GenieMaster architecture.** EEP connects four contracts: Capability (1), Guidance (2), Feedback (3), and Continuity (4). Guidance and Feedback form the bidirectional agent–policy interface; co-designed annotations and training make guidance usable by the policy. Task context retains ongoing work, while reflection and experience memory adapt subsequent decisions.

observation and recovery. New domains compose supported capabilities, and new embodiments implement their execution interfaces, subject to the target robot's capabilities and operating conditions.

Guiding execution and interpreting feedback. For learned-policy execution, the **Guidance** and **Feedback contracts** form the bidirectional agent–policy interface. The agent supplies a subtask instruction and optional spatial cues, detailed language, or object references. The policy acts on current observations under this guidance; Section 3.2 explains how training makes it usable.

An invocation's end alone does not establish physical success. An asynchronous success/error detector assesses current and recent observations against agent-specified completion criteria and error signals, returning a state estimate and evidence. Two consecutive `done` assessments within an invocation authorize termination and progression (Appendix E). Recoverable local errors prompt revised guidance and another invocation; failures that invalidate downstream prerequisites prompt revision of the remaining action list.

Preserving task continuity. The harness retains assignments, confirmed progress, dependencies, and unfinished work under the **Continuity contract**. Combined with current evidence, this state supports resumption: in hosting guests, two robots suspend activities to retrieve a towel together, then return to them (Appendix A.1). Incoming store tasks are assigned to available, capable robots (Section 4.7). Task-level coordination leaves tightly synchronized motion to local controllers or coordinated policies. Task context preserves ongoing work; experience memory retains reusable lessons for later decisions.

3.2 Agent–Policy Co-Design for In-Context Guidance

To implement EEP's Guidance contract, we train policies on demonstrations paired with the conditions the agent supplies at inference. The guidance interface, demonstration annotations, and training inputs are designed together. The coding agent and policy are not jointly trained.

The policy receives current observations o_t, robot state s_t, the subtask instruction ℓ, and the latest available guidance g_t:

$$\hat{a}_t \sim \pi_\theta(\cdot \mid o_t, s_t, \ell, g_t)\,, \tag{1}$$

where $\hat{a}_t$ is a predicted action or action chunk. Guidance persists between agent updates while observations change; training pairs these inputs with demonstrated actions a_t. Visual guidance is supplied through additional image inputs alongside the usual camera views.

Spatial guidance. Bounding boxes identify target regions; points specify grasp, contact, or placement locations. We restrict spatial guidance to bounding boxes and points, as agent-generated trajectories were insufficiently accurate in our tests.

During deployment, the agent annotates a previously captured head-camera image, which remains a fixed guidance input as the policy receives new observations. We account for this temporal offset through **temporal-offset augmentation** during policy training. Bounding-box and point annotations on sampled keyframes are extended to per-frame labels using tracking models, including SAM 3 (Carion et al., 2025). For each training example, we sample a guidance frame between the start of the current subtask and the current observation, overlay its spatial annotations, and pair the annotated frame with the current observation and target action (Appendix B).

Visual goal conditioning. An object reference specifies target appearance, e.g., "Pick the item shown in `<image>`." The agent retrieves it from local assets or obtains it through web search or image generation. Training pairs references with observations and actions for the corresponding targets: the reference specifies what to manipulate, while current observations show where to act. The agent prioritizes a suitable object reference for target selection and otherwise generates spatial guidance; the two visual conditions need not be combined (Figure 5).

DensePrompt. Optional fine-grained language specifies execution details, e.g., "Approach and grasp the object from the right at roughly $45°$." Demonstrations pair such descriptions with the corresponding actions, teaching the policy to follow approach, grasp, or sequence instructions beyond the subtask goal.

3.3 Continual Harness Adaptation

Harness evolution updates the context used to guide, assess, and orchestrate execution, with model parameters fixed during deployment.

Temporary and long-term memory. Temporary memory stores textual execution records: task objectives, observation summaries, detector outputs, and success or failure outcomes. For failures, GenieMaster automatically appends summaries of likely causes. Long-term memory retains reusable guidance, assessment criteria, recovery steps, and coordination strategies across tasks.

Reflection and memory consolidation. Periodically or on demand, GenieMaster asynchronously reviews temporary records across robots, assessing execution outcomes and detector judgments against task objectives and observations. This reflection runs outside online monitoring and consolidates the resulting lessons into long-term memory. The agent uses these instructions to guide subsequent execution, assess completion, and coordinate work across compatible robots. Appendix D traces this process through execution records and memory updates between office-supply search rounds.

4 Experiments

We evaluate three questions: **(1)** Does the complete harness improve end-to-end execution? **(2)** Does reference-conditioned training improve retrieval of unseen products? **(3)** How does execution experience inform later behavior? System-level comparisons assess the complete configuration; the paired product study tests reference conditioning with matched demonstrations and training settings.

Table 1: GENIEMASTER performance across six real-world tasks. Success rate (SR) and mean task completion (TC) are scored independently and reported as percentages. Each method is evaluated on 100 trials with randomized layouts per task. Overall is the unweighted mean across the six tasks. D1–D3 correspond to hosting guests, tablecloth spreading, and office-supply search.

Method	Metric	D1	D2	D3	Item Retrieval	Restocking	Cleaning	Overall
Vanilla VLA	SR	0	15	0	20	53	25	18.83
	TC	48.20	58.33	18.05	81.16	62.50	65.83	55.68
Dual-system	SR	18	25	32	40	58	60	38.83
	TC	68.20	52.00	21.05	80.55	67.76	78.30	61.31
GenieMaster	SR	45	60	84	93	95	95	78.67
	TC	84.54	76.75	89.80	96.67	96.05	94.44	89.71
Human Orchestration	SR	50	55	95	95	96	95	81.00
	TC	86.60	76.45	92.11	98.74	98.33	94.79	91.17

4.1 EXPERIMENTAL SETUP

We evaluate six tasks on AGIBOT G1, AGIBOT G2, and ARX AC ONE: hosting guests, tablecloth spreading, office-supply search, item retrieval, restocking, and cleaning. The last three take place in a convenience store with over 1,000 products. Appendix A details the configurations and procedures.

The overall store evaluation includes policy-driven navigation and manipulation for all four methods. SLAM navigation APIs are supported by the harness but unused in this comparison.

Unless otherwise stated, GENIEMASTER uses Codex with GPT-5.6 Sol, policies fine-tuned from $\pi_{0.5}$, and Qwen3.8-VL-27B for success/error detection. All policies trained for our experiments use demonstration datasets of equal total duration (Appendix B). Outside the experience studies, trials start from the same initial memory; task state and temporary experience persist within each trial.

4.2 OVERALL END-TO-END PERFORMANCE

Table 1 compares four execution workflows under matched policy-training conditions:

- **Vanilla VLA:** Executes one whole-task instruction, without decomposition, instruction switching, or auxiliary guidance.
- **Dual-system:** A fine-tuned Qwen3.8-VL-27B generates and updates subtask instructions from observations during execution. The policy follows these instructions without auxiliary guidance.
- **GENIEMASTER:** A coding agent organizes tasks and guides policy execution. Success/error detection reports physical outcomes, which the agent combines with retained task state to advance, recover, or revise the plan.
- **Human Orchestration:** Humans decompose tasks, assess completion, and switch instructions, using the same policies, guidance modalities, and tools as GENIEMASTER.

All policy variants in this comparison are fine-tuned from the same base model on the same demonstration data, with matched training steps, learning rates, and other training settings. Variants are trained separately for their input and execution configurations. Qwen3.8-VL-27B serves different roles in the two automated workflows: subtask generation in Dual-system and success/error detection in GENIEMASTER. Across all tasks, GENIEMASTER supplies subtask instructions; additional guidance is selected per invocation from the task-specific options in Appendix A.

Evaluation metrics. End-to-end success rate (SR) is the percentage of trials achieving the complete task goal. For each trial, task completion (TC) averages process scores over predefined, equally

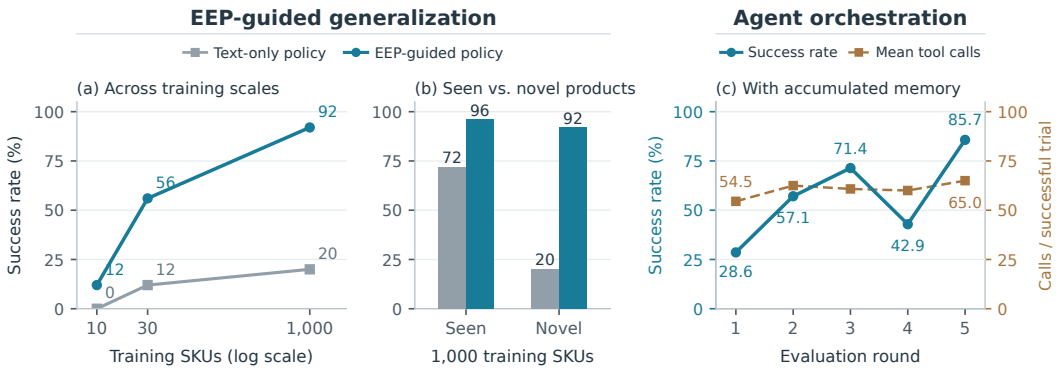


Figure 3: **Generalization and experience-based orchestration. (a)–(b)** Separately trained text-only and EEP-guided policies, with object references supplied to the latter: novel-product retrieval across training SKU counts (log scale), and seen versus novel products at 1,000 SKUs. Each split has 25 trials per policy, without navigation. **(c)** Office-supply search over five rounds of seven trials: success over all trials and mean tool calls over successful trials only.

weighted subgoals; flawed completion receives 0.5, for example after a clear collision or repeated attempts. Later state changes do not reduce earned scores. Human evaluators score SR and TC separately based on observed physical outcomes. Table 1 reports SR and mean TC as percentages over the same trials, including failures.

Recovery and retries are permitted for recoverable execution errors. A trial is terminated and scored as unsuccessful for SR if the robot enters an unrecoverable state, selects the wrong object, places an object at an incorrect destination, or requires human physical assistance. Task descriptions and execution configurations appear in Appendix A.

GENIEMASTER achieves 78.67% macro-averaged SR and 89.71% TC, versus Dual-system's 38.83% and 61.31%, exceeding both automated baselines on every task (Table 1). The gains show the effectiveness of the complete GENIEMASTER workflow under matched policy-training conditions. Section 4.4 examines object-reference conditioning in a paired comparison.

With the same policies and tools, GENIEMASTER approaches Human Orchestration overall and attains higher observed SR and TC on D2 (tablecloth spreading).

4.3 CAPABILITY COMPOSITION AND CONTROL ALLOCATION

We examine capability composition and guidance within EEP through five cumulative configurations for fixed-location retrieval, with matched product types and layout and no navigation (Table 2). All five configurations use GPT-6 Astra as the agent model. C3–C5 share one policy checkpoint; C5 also changes head/waist control allocation.

Table 2: Cumulative capabilities and retrieval performance over 100 trials with randomized layouts per configuration. Checkmarks denote available tools; C5 transfers head/waist control to the agent. Time is the mean over successful trials, including retries.

Config.	Agent arm control (IK)	Grasp planning	Guided policy	Object reference	Agent head/ waist control	SR (%)	Time (s)
C1	✓	–	–	–	–	25	2,143
C2	✓	✓	–	–	–	40	681
C3	✓	✓	✓	–	–	83	32
C4	✓	✓	✓	✓	–	94	16
C5	✓	✓	✓	✓	✓	**97**	14

Configurations. In C1, the agent directly controls the arm through an inverse-kinematics interface; C2 adds AnyGrasp (Fang et al., 2022) grasp proposals and cuRobo (Sundaralingam et al., 2023) motion planning. C3 introduces the guidance-trained policy using agent-drawn boxes on historical observations. C4 supplies catalog object-reference images for visual goal conditioning, prioritized over online box generation; the two visual conditions need not be combined. C5 disables the policy's head/waist actions and exposes direct head/waist control to the agent.

Results. Introducing the guidance-trained policy with spatial cues raises SR from 40% to 83%; adding object references brings it to 94%. With head and waist free to move, policy control exhibited repeated hesitation and turning back and forth. Assigning their control to the agent raises SR to 97%. Mean retrieval times are reported over successful trials only, including successes after retries. Because the successful-trial subsets differ, their ratios do not establish overall speedup. This control transfer illustrates configurable agent–policy responsibilities within the same harness. These cumulative configurations evaluate tool composition and control allocation, not isolated component effects.

Temporal alignment. With the same training data and otherwise identical training settings, removing temporal-offset augmentation reduces convenience-store item-retrieval success by 9 percentage points. This ablation supports varying the age of spatial reference frames during training to match delayed and retained guidance at deployment (Appendix B).

4.4 Unseen-Product Generalization through In-Context Guidance

We evaluate guidance–policy co-design by comparing separately trained text-only and EEP-guided policies on held-out products. Both receive observations, robot state, and the task instruction; the EEP-guided policy additionally receives object references during training and agent-supplied references at deployment. At each training scale, action demonstrations, training steps, and batch size are matched; only object-reference conditioning and its inputs differ. The comparison evaluates reference-conditioned training together with inference-time guidance.

Retrieval is evaluated at a fixed location without navigation. At each fine-tuning scale (10, 30, and 1,000 SKUs), each policy receives 25 trials on held-out product identities. At 1,000 SKUs, we additionally test 25 seen products, with one trial per identity in each split.

At 1,000 training SKUs, the EEP-guided policy succeeds on 23/25 novel products (92%) versus 5/25 (20%) for the text-only policy; seen-product results are 24/25 (96%) and 18/25 (72%) (Figure 3(b)). The seen–novel success gap narrows from 52 to 4 percentage points.

Across training scales of 10, 30, and 1,000 SKUs, novel-product success is 12%, 56%, and 92% for EEP-guided policies, versus 0%, 12%, and 20% for text-only policies (Figure 3(a)). These results support using learned reference-conditioned retrieval for product identities outside the training SKU set.

4.5 Learning from Experience

Completion assessment. We examine whether reflection on execution experience can improve subtask completion assessment by refining the detector's context. An auxiliary comparison tests seven completion detectors before and after one round of context reflection with parameters fixed. Completion-boundary alignment improves for six and remains unchanged for one; this measures assessment quality, not online skill-switching latency (Appendix E).

Orchestration. We use the office-supply search task to evaluate whether the harness can autonomously improve its orchestration through experience, running five rounds of seven trials each. Memory is consolidated after each round rather than daily; temporary memory is then cleared and long-term memory retained. We measure success over all trials and mean tool calls over successful trials.

Success rises from 28.6% in the first round to 85.7% in the fifth, accompanied by only a modest increase in tool calls per successful trial. Figure 3(c) shows a marked setback in round four, when success drops from 71.4% to 42.9% because of incorrect information incorporated into long-term memory after round three. During round four, GenieMaster autonomously corrects these

Table 3: LIBERO-Pro success (%) and gains Δ (percentage points). FT data: policy fine-tuning datasets; L40/L90: LIBERO-40/LIBERO-90. *: six conditions, otherwise eight.

(a) Comparison with policy baselines

Method	FT data	Success
OpenVLA	L40	0
π_0	L40	0
X-VLA	L40	3.8
$\pi_{0.5}$	L40	10.75
π_{RLinf}	L40 + L90	50.0
GENIEMASTER	L40	**60.38**

(b) Harness systems and gains

Method	FT data	Baseline	Success	Δ
CaP-X*	–	–	18.2	–
Harness VLA (Codex)	L40 + L90	50.0	72.1	+22.1
Harness VLA (CC)	L40 + L90	50.0	**82.4**	+32.4
Pigey*	L40	12.8	53.3	+40.5
GENIEMASTER	L40	10.75	60.38	**+49.6**

memory entries, enabling success to rebound in round five. After an earphone-case misidentification, memory adds its silver-white, round appearance and close-range verification under occlusion. These instructions are retained in the consolidated memory shown in Appendix D.

4.6 EVALUATION ON LIBERO-PRO

We evaluate LIBERO-Pro (Zhou et al., 2025) on Spatial, Object, Goal, and LIBERO-10 under position-swap and task perturbations, with ten trials per task and condition. Our $\pi_{0.5}$ policy is fine-tuned on LIBERO-40 without LIBERO-90 (Liu et al., 2023). GENIEMASTER reaches 60.38% success, exceeding the listed policy baselines (Table 3, left). Its 49.6-point gain is largest among the compared harnesses (right), a ranking retained on six shared conditions (Appendix F). The gain includes training and agent execution, as our guided checkpoint differs from the baseline.

GENIEMASTER averages 22.4 s per execution, combining task performance with responsiveness for real-world deployment. Harness VLA achieves higher success, but its separate Codex study reports mean planning times of 344.17–529.23 s per episode (Zhang et al., 2026b).

4.7 PERSISTENT DEPLOYMENT IN A CONVENIENCE STORE

We deploy GENIEMASTER in a real convenience store over a 24-hour window, where scheduled inspections identify follow-up work and incoming customer orders initiate retrieval tasks. The agent assigns available, capable robots and uses physical feedback and retained task state to coordinate ongoing execution. Appendix G presents retrieval, restocking, and cleaning cases with observations, EEP-guided instructions, visual guidance, and execution results. Task-level evaluations above quantify performance.

5 LIMITATIONS

GENIEMASTER depends on policy competence and guidance quality. False success can advance work prematurely; conservative detection delays it. Tightly synchronized manipulation requires local controllers or coordinated policies. Spurious or stale memories motivate provenance tracking, regression checks, and rollback; non-monotonic office-supply search results caution against assuming steady gains from experience. Further evaluation across a broader range of tasks, environments, and robot platforms would help assess the generality of these findings.

6 CONCLUSION

GENIEMASTER connects frontier-agent reasoning to responsive robot execution for persistent multi-robot work. Its bidirectional EEP connects capability invocation and policy guidance to physical feedback. Co-designed guidance and policy training make agent-acquired context usable during manipulation; retained task state supports coordination and resumption; reflection and memory inform later decisions. These results mark a step toward more robust and general-purpose robotic systems.

AI USE STATEMENT

We used generative AI tools to assist with manuscript organization and language editing. The authors reviewed the AI-assisted revisions and take responsibility for the final text, claims, and reported results.

REFERENCES

Michael Ahn et al. Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. URL https://arxiv.org/abs/2204.01691.

Suneel Belkhale et al. RT-H: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024. URL https://arxiv.org/abs/2403.01823.

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

Kevin Black et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL https://arxiv.org/abs/2307.15818.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. URL https://arxiv.org/abs/2511.16719.

Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. RACER: Rich language-guided failure recovery policies for imitation learning. *arXiv preprint arXiv:2409.14674*, 2024. URL https://arxiv.org/abs/2409.14674.

Xin Ding, Liang Mi, Mingzhe Huang, Zixuan Wang, Chao Zhang, Zixu Hao, Fu Chen, Xiangyu Li, Yikai Zheng, Yaoyu Guo, Weijun Wang, Kun Li, Hao Wu, Yunxin Liu, and Ting Cao. Zetta ζ: An efficient closed-loop embodied harness for self-evolving physical intelligence. *arXiv preprint arXiv:2608.16590*, 2026. URL https://arxiv.org/abs/2608.16590.

Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. AnyGrasp: Robust and efficient grasp perception in spatial and temporal domains. *arXiv preprint arXiv:2212.08333*, 2022. URL https://arxiv.org/abs/2212.08333.

Letian Fu, Justin Yu, Karim El-Refai, Ethan Kou, Haoru Xue, Huang Huang, Wenli Xiao, Guanzhi Wang, Dantong Niu, Fei-Fei Li, Guanya Shi, Jiajun Wu, Shankar Sastry, Yuke Zhu, Ken Goldberg, and Linxi Fan. CaP-X: A framework for benchmarking and improving coding agents for robot manipulation. *arXiv preprint arXiv:2603.22435*, 2026. URL https://arxiv.org/abs/2603.22435.

Liane Galanti, Dhruv Shah, and Tri Dao. Addressing the orchestration gap in generalist robots via physical agency. *arXiv preprint arXiv:2607.21725*, 2026. URL https://arxiv.org/abs/2607.21725.

Nick Hawes et al. The STRANDS project: Long-term autonomy in everyday environments. *arXiv preprint arXiv:1604.04384*, 2016. URL https://arxiv.org/abs/1604.04384.

Jiaheng Hu, Mohit Shridhar, Caden Lu, Dhruv Shah, Hao-Tien Lewis Chiang, Jie Tan, and Annie Xie. What matters in orchestrating robot policies: A systematic study of hierarchical VLA agents. *arXiv preprint arXiv:2606.10267*, 2026. URL https://arxiv.org/abs/2606.10267.

Jinbang Huang, Yuanzhao Hu, Zhiyuan Li, Ran Qi, Yixin Xiao, et al. RoboHarness: Memory-driven orchestration of heterogeneous robot policies for long-horizon planning. *arXiv preprint arXiv:2607.18060*, 2026. URL https://arxiv.org/abs/2607.18060.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Fei-Fei Li. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. URL https://arxiv.org/abs/2409.01652.

Wenlong Huang et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022. URL https://arxiv.org/abs/2207.05608.

MoniJesu James, Marcelino Julio Fernando, Miguel Altamirano Cabrera, and Dzmitry Tsetserukou. AnyGoal: Vision-language guided multi-agent exploration for training-free lifelong navigation. *arXiv preprint arXiv:2606.13878*, 2026. URL https://arxiv.org/abs/2606.13878.

Weiqi Jin, Peijun Tang, Kuncheng Luo, Baifu Huang, Binyan Sun, Haotian Yang, Shangjin Xie, and Jianan Wang. A glimpse into long-term physical coexistence with intelligent robots. *arXiv preprint arXiv:2607.11377*, 2026. URL https://arxiv.org/abs/2607.11377.

Borui Kang, Jinrui Gu, Junhan Lv, Wenbin Li, Lei Wang, and Yang Gao. Harness continual learning: Continual adaptation beyond model parameters. *arXiv preprint arXiv:2608.19013*, 2026. URL https://arxiv.org/abs/2608.19013.

Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. SMART-LLM: Smart multi-agent robot task planning using large language models. *arXiv preprint arXiv:2309.10062*, 2023. URL https://arxiv.org/abs/2309.10062.

Seth Karten, Joel Zhang, Tersoo Upaa, Jr, Ruirong Feng, Wenzhe Li, Chengshuai Shi, Chi Jin, and Kiran Vodrahalli. Continual harness: Online adaptation for self-improving foundation agents. *arXiv preprint arXiv:2605.09998*, 2026. URL https://arxiv.org/abs/2605.09998.

Moo Jin Kim et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Xunyao Lei, Renjun Wu, Tianlin Huo, and Xuesong Li. SparkVLA: Stop-aware hierarchical VLA with adaptive action chunking for long-horizon manipulation. *arXiv preprint arXiv:2608.16172*, 2026. URL https://arxiv.org/abs/2608.16172.

Yi Li et al. HAMSTER: Hierarchical action models for open-world robot manipulation. *arXiv preprint arXiv:2502.05485*, 2025. URL https://arxiv.org/abs/2502.05485.

Jacky Liang et al. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022. URL https://arxiv.org/abs/2209.07753.

Yiran Ling, Qing Lian, Jinghang Li, Qing Jiang, Tianming Zhang, Xiaoke Jiang, Chuanxiu Liu, Jie Liu, and Lei Zhang. Guide, think, act: Interactive embodied reasoning in vision-language-action models. *arXiv preprint arXiv:2605.13632*, 2026. URL https://arxiv.org/abs/2605.13632.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023. URL https://arxiv.org/abs/2306.03310.

Model Context Protocol Contributors. Tools: Model Context Protocol specification. Online specification, 2025. URL https://modelcontextprotocol.io/specification/2025-11-25/server/tools. Accessed September 17, 2026.

Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. OmniManip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. *arXiv preprint arXiv:2501.03841*, 2025. URL https://arxiv.org/abs/2501.03841.

Physical Intelligence et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Physical Intelligence et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026. URL https://arxiv.org/abs/2604.15483.

Ke Rui, Yushen Zuo, Jiawei Wang, Haoran Jia, Jinming Ma, Weitao Zhou, and Minglei Li. Diagnosing semantic handoff failures in agent-orchestrated vision-language-action skill composition. *arXiv preprint arXiv:2607.06256*, 2026. URL https://arxiv.org/abs/2607.06256.

Lucy Xiaoyang Shi et al. Hi Robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025. URL https://arxiv.org/abs/2502.19417.

Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. cuRobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023. URL https://arxiv.org/abs/2310.17274.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. URL https://arxiv.org/abs/2305.16291.

Kailin Wang, Haoxiang Jie, Yaoyuan Yan, Zhiyou Heng, and Zhaosong Li. RegenHarness: A robot agent harness with evidence-gated recursive self-improvement. *arXiv preprint arXiv:2609.27612*, 2026a. URL https://arxiv.org/abs/2609.27612.

Qi Wang, Tianyi Wang, Chengyang Li, Shikun Ban, Yurun Chen, Yizhong Ge, Jason Qin, Chengtai Li, and Wentao Zhu. Towards the harness of embodied agents. *arXiv preprint arXiv:2608.11246*, 2026b. URL https://arxiv.org/abs/2608.11246.

Zixuan Wang, Yuxin Chen, Yuqi Liu, Jinhui Ye, Pengguang Chen, Changsheng Lu, Shu Liu, Bei Yu, and Jiaya Jia. VP-VLA: Visual prompting as an interface for vision-language-action models. *arXiv preprint arXiv:2603.22003*, 2026c. URL https://arxiv.org/abs/2603.22003.

Sihyung Yoon, Minjong Yoo, Sanghyun Ahn, Seojeong Choi, and Honguk Woo. RoboBRIDGE: A modular framework for bridging policies to robust real-world robotic agents. *arXiv preprint arXiv:2607.27881*, 2026. URL https://arxiv.org/abs/2607.27881.

Gehao Zhang, Zhenyang Ni, Payal Mohapatra, Han Liu, Ruohan Zhang, and Qi Zhu. EmboAlign: Aligning video generation with compositional constraints for zero-shot manipulation. *arXiv preprint arXiv:2603.05757*, 2026a. URL https://arxiv.org/abs/2603.05757.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, et al. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026b. URL https://arxiv.org/abs/2607.08448.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025. URL https://arxiv.org/abs/2510.10274.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025. URL https://arxiv.org/abs/2510.03827.