LEARNING FROM PHYSICAL EXPERIENCE: CONTEXT-SPACE ADAPTATION FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent multimodal foundation models exhibit strong perception, reasoning, and in-context learning capabilities, but translating these capabilities into physical action remains challenging. Robotic manipulation demands four critical capabilities: spatial reasoning, dexterous interaction, tool use, and adaptation to execution failures. Through real-world experiments across diverse manipulation tasks, we find that ChatGPT-6-Astra, a state-of-the-art general-purpose reasoner, can serve as a physically actionable, context-responsive manipulation policy, demonstrating effective fine manipulation and spatial reasoning.[1] Building on these findings, we introduce a recursive self-improvement harness for learning from physical experience. A context compiler integrates current observations, system-level guidance, task instructions, and retrieved experience to generate subtask sequences and executable motion plans. Plans, execution results, and task outcomes are accumulated as experience, with optional human feedback providing additional guidance. Reflection on this evidence produces context updates that inform subsequent planning and execution, forming a recursive loop of interaction, experience accumulation, and context adaptation without task-specific parameter updates. Our experiments show that learning from experience enables manipulation adaptation in context space. On challenging tool-use and potato-chip picking tasks where direct reasoning alone yields zero success, successive experience-driven updates support progressive skill consolidation and improved execution. These findings connect general-purpose reasoning with physical skill acquisition and demonstrate how accumulated interaction experience can improve robotic manipulation while keeping model parameters fixed.

1 INTRODUCTION

A long-standing goal of robotic manipulation is to perform diverse tasks reliably in unfamiliar and changing environments. The challenge is to turn limited physical experience into reusable improvements. Vision–language–action (VLA) models specialize through robot demonstrations and supervised fine-tuning (Kim et al., 2024; Physical Intelligence et al., 2025b). DreamZero jointly predicts video and actions for zero-shot execution, while Cosmos Policy uses demonstration-based fine-tuning and rollout-informed planning (Ye et al., 2026; Kim et al., 2026). Real-world reinforcement learning improves policy parameters through physical interaction, requiring reward specification and scene reset or recovery infrastructure (Physical Intelligence et al., 2025a; Luo et al., 2024; 2025). General-purpose multimodal reasoners offer a complementary interface: observations, instructions and execution feedback can condition executable actions without updating model weights. Their relevance depends on whether contextual flexibility supports reusable improvement under a limited physical interaction budget.

We investigate this possibility with GPT-6 Astra (OpenAI, 2026) across fine-object pickup, bottle reorientation, tong-mediated bread transfer, bimanual sweeping, towel folding, drawer search and fragile-object handling. Two findings emerge: Astra can execute precision-demanding manipulation

[1] Detailed analysis can be found in https://submission-56861.speedy-grove-4214.chatgpt.site/.

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.

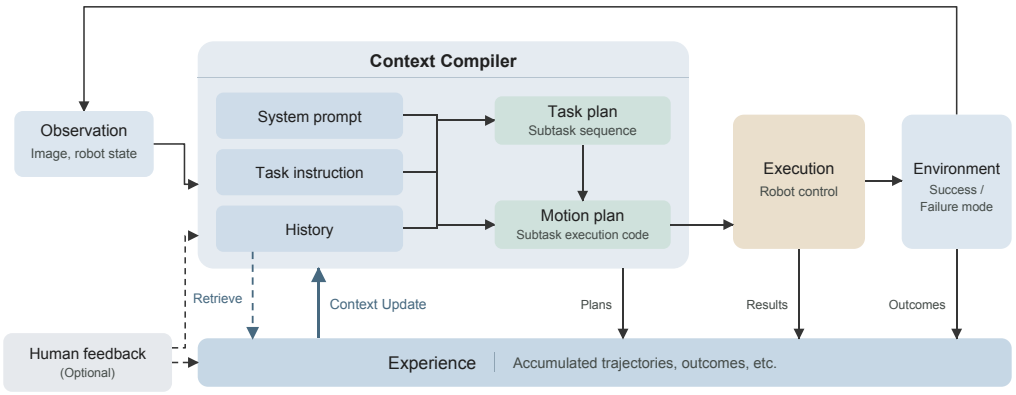

Figure 1: **From physical execution to recursive context-space adaptation.** The context compiler integrates current observations, a system prompt, task instructions, and retrieved experience to generate a task plan and executable motion code. Robot execution changes the environment, producing new observations for subsequent execution, accumulating as experience that informs context updates for later attempts without parameter changes.

and spatially coordinated actions, and its behavior is steerable through visual feedback, recovery demonstrations and language corrections. History-enabled conditions show higher observed success on several tasks; towel-folding and barbecue-transfer cases show revisions to grasp checking, repositioning, motion direction, arm selection and action order (Section 2.2). These findings establish prerequisites for experience-derived context adaptation.

Building on these observations, we introduce FreeMani, a recursive self-improvement harness that connects physical execution, reflection and context adaptation (Figure 1). A context compiler combines current observations with configurable sources of history, demonstrations and extracted experience to condition executable actions. After execution, a separate reflection step examines the action, its execution outcome and the resulting scene, distilling potentially reusable evidence into structured experience: an applicability condition, a tentative interpretation, an actionable correction and supporting outcome records. This experience can then inform subsequent decisions, whose physical consequences provide new evidence for further revision. The actor and reflection models remain frozen, and the action interface stays fixed; recursion operates through the behavior–experience–context loop. Separating raw history from extracted experience, with bounded retrieval when enabled, makes the adaptation mechanism explicit and its information flow auditable. The design builds on reflective context optimization (Agrawal et al., 2025; Cai et al., 2025) and prior physical adaptation systems (Lu et al., 2026; Xiao et al., 2026; Fu et al., 2026; Zhang et al., 2026b).

Our evaluation connects executable competence, context responsiveness and interaction-driven exploration with local reuse. Request-level audits also show that a nominal history or experience condition does not guarantee delivery to the actor. We distinguish observed recovery and within-session reuse from retained learning: the latter requires comparing no persistent correction, expert-supplied correction and model-extracted correction after resetting both scene and conversation.

2 Problem Statement

2.1 Objective: Improve Policy Behavior Through Reusable Experience

Our objective is to improve manipulation by converting physical experience into reusable guidance. We move the adaptation variables from model parameters to the context conditioning a frozen policy. Context can change subtask selection, approach geometry, grasp verification and failure response. Following Training-Free GRPO, successful and unsuccessful attempts provide natural-language guidance for subsequent decisions (Cai et al., 2025).

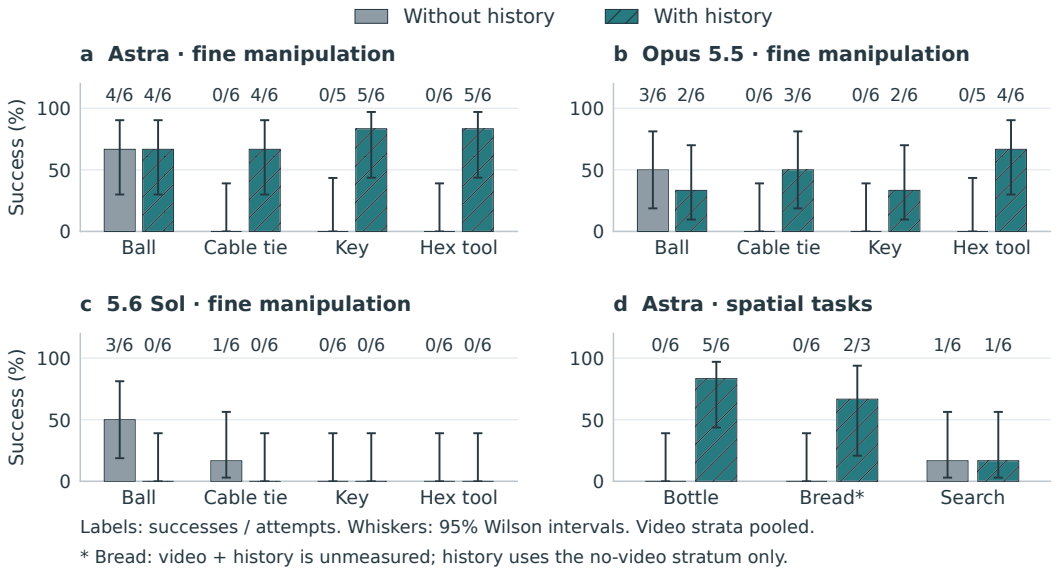

Figure 2: Preliminary capability and context-use evidence (set D). Bars pool counts over video; whiskers are descriptive 95% Wilson intervals. Astra's observed rate is higher on three fine tasks and bottle standing, but unchanged on ball and search. *Bread lacks video+history; its matched no-video contrast is 0/3 versus 2/3. These comparisons motivate learning experiments; they do not measure persistent learning.

From execution records to reusable experience. An attempt produces observations, task and motion plans, actions, tool responses and outcomes, with recorded rationales or reasoning summaries. Successful attempts reveal strategies worth reusing; failures expose discrepancies between intended and observed effects. The experience store connects each situation to an actionable recommendation and an observable consequence to check, retaining links to supporting images and records. Entries can be added, modified, deleted or retained as evidence accumulates (Cai et al., 2025; Zhao et al., 2023).

From experience to context-conditioned behavior. The *Context Compiler* assembles observations, system guidance, task instructions, interaction history and relevant stored experience (Figure 1). The actor produces a subtask sequence and executable motion code. History supports the current attempt; stored experience provides guidance for subsequent attempts. Optional human feedback redirects behavior and supplies corrective evidence. Execution yields new observations and outcomes for further experience revision.

The objective is reusable behavioral improvement. Context-space adaptation aims to increase success, reduce repeated failures and improve efficiency. Recovery can also benefit from a favorable state left by earlier actions, so retained improvement must be tested after scene and conversation resets. Recursion refers to repeated execution and experience revision with fixed weights.

2.2 Preliminary capability and context-use evidence of GPT-6 Astra

We vary video access and interaction history, and examine recovery demonstrations and online language feedback to test three prerequisites: fine manipulation, spatial reasoning and context-sensitive strategy revision.

Fine manipulation benefits substantially from memory guidance. Figure 2a compares Astra's small-object pickup performance with and without interaction history, pooling the available video conditions. Overall success rises from $4/23$ (17.4%) without history to $18/24$ (75.0%) with history. The task-level contrasts are $0/6$ versus $4/6$ for cable ties, $0/5$ versus $5/6$ for keys, and $0/6$ versus $5/6$ for the hex tool. Ball pickup remains $4/6$ in both conditions.

These contrasts show executable fine manipulation with actor-dependent history benefits: Opus 5.5 improves from $3/23$ to $11/24$, whereas 5.6 Sol changes from $4/24$ to $0/24$ (Figure 2b–c). Which history contents produce the gains remains open.

Spatial reasoning supports reorientation and occasional successful search. Astra stands a fallen bottle upright in $5/6$ history-enabled trials versus $0/6$ without history, including $3/3$ under video+history. In the wrist-camera recovery case described by the authors, it succeeds despite severe visual occlusion, providing an example of spatially coordinated action under partial visibility. Tong-mediated bread transfer further combines spatial positioning with tool use: the matched no-video comparison is $0/3$ without history versus $2/3$ with history.

Drawer-based card search succeeds in only $1/6$ trials both with and without history; drawer-handle grasping is a frequent obstacle in the experimental log. Spatial reasoning thus remains dependent on perception and successful manipulation, and history provides no observed search benefit.

Context steering changes strategies and follows spatially specific instructions. Figure 3a compares towel folding under clean and recovery demonstrations. With the clean demonstration, Astra grasps the near corners, retries the fold, and gives up at C24. With recovery context, it verifies edge capture, maintains the grasp, brings the towel closer, and inspects the final fold at C26. The difference concerns grasp verification and workspace management, showing that demonstration context can redirect the manipulation strategy.

Figure 3b illustrates online instruction following during barbecue transfer. A request to move backward is followed by a 4.4 cm displacement; feedback requesting more clearance produces a 12.8 cm lift; and guidance to clear the left arm produces a 12.0 cm left-arm motion while the right remains fixed. "Up, then right" guidance leads to a 2.7 cm rise followed by a 4.2 cm lateral motion before the plan is superseded. These examples demonstrate responsiveness to motion direction, arm selection, and action order. Visual demonstration context and textual feedback thus provide concrete channels for revising both high-level strategies and individual actions.

From context steering to learning. These results establish executable capability and a context channel that can redirect behavior. Learning requires the system to extract guidance from its own attempts and retain its benefit in new executions. Section 4 examines exploration and local reuse; fresh-reset retention remains untested.

3 A RECURSIVE SELF-IMPROVEMENT HARNESS

Can a frozen actor turn evidence from its own physical interaction into useful context? FreeMani connects action, observation, reflection and context update. At each step, the actor receives current observations and configurable auxiliary sources, then produces executable actions. A separate frozen reflection model distills execution evidence into structured experience for subsequent decisions.

At decision step t, let o_t denote the current multimodal observation and let $\mathcal{S}_t$ denote the set of auxiliary context sources. The actor selects

$$a_t \sim \pi_\theta \left(a \mid \mathrm{Compile}(o_t; \mathcal{S}_t), C\right), \tag{1}$$

where θ denotes the frozen multimodal model, C is the fixed action contract, and $\mathrm{Compile}(\cdot)$ constructs the model request. $\mathcal{S}_t$ can contain recent interaction history, captured experience, demonstrations, or other task context.

Self-improvement operates through context adaptation: behavior generates evidence, evidence becomes experience, and experience can change subsequent behavior.

3.1 PHYSICAL INTERACTION AS LEARNING EVIDENCE

Physical execution provides a source of information that is unavailable from static visual reasoning alone. After the actor selects an action a_t, execution produces a new observation

$$o_{t+1} \sim P(\cdot \mid o_t, a_t), \tag{2}$$

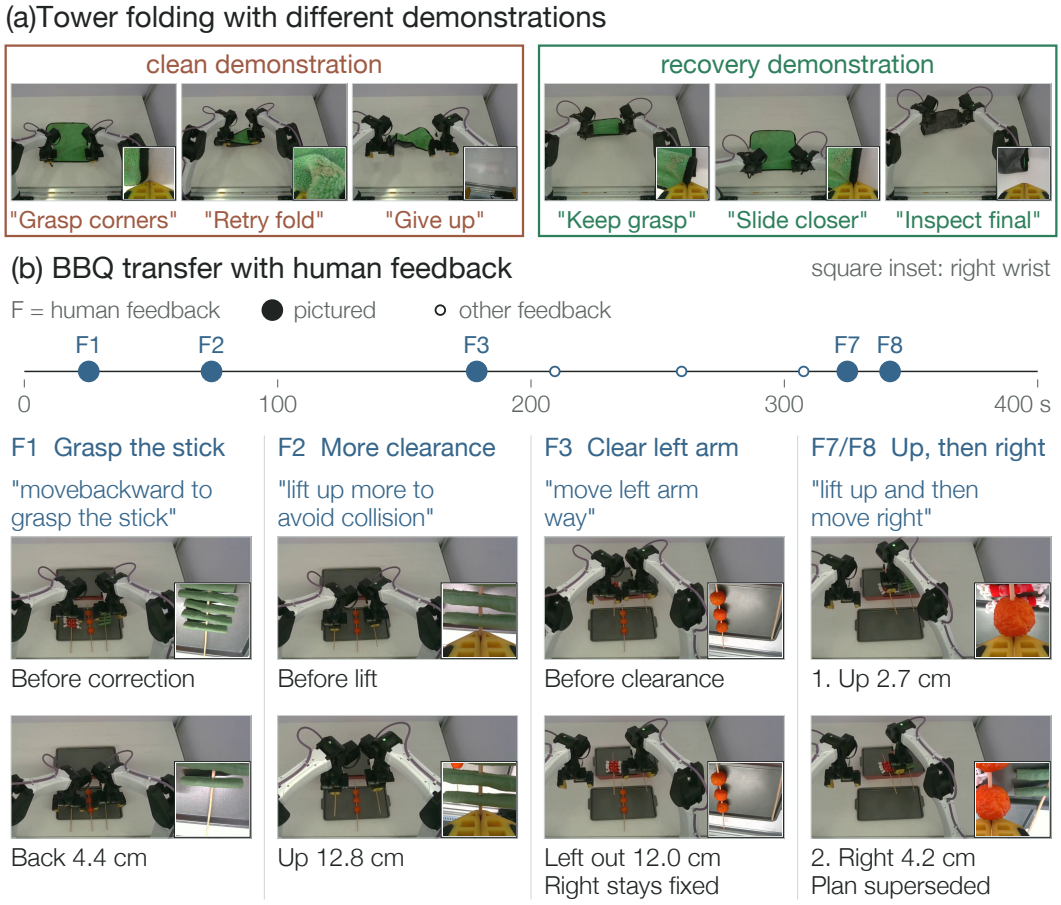

Figure 3: Context steering and instruction following in robotic manipulation. (a) Clean and recovery demonstrations lead to different grasp-checking and repositioning strategies during towel folding. (b) Online language feedback during barbecue transfer is followed by direction-, arm-, and order-specific motion revisions. Photographs, feedback indices, and displacement annotations are author-supplied selected evidence; these cases illustrate behavioral responsiveness rather than a repeated estimate of feedback-induced success gains.

together with an execution outcome y_t. We represent the resulting interaction evidence as

$$r_t = (a_t, y_t, o_{t+1}). \tag{3}$$

Here, y_t records interface outcomes and o_{t+1} the physical scene. An action may execute without achieving its intended effect; a rejected action may reveal a strategy constraint. This action–consequence transition is the basic unit of experience extraction, while action revision alone does not establish improvement.

3.2 From Interaction to Structured Experience

Raw history records what happened; extracted experience identifies what should change or repeat in a similar situation.

For each completed interaction, a frozen reflection model examines the available execution evidence and determines whether it contains a reusable lesson. Not every interaction needs to produce an update:

$$e_t = \begin{cases} U_\theta(r_t), & \text{if } r_t \text{ contains reusable evidence,} \\ \varnothing, & \text{otherwise,} \end{cases} \tag{4}$$

where U_θ denotes the experience extractor. A retained entry contains an applicability condition, tentative explanation, actionable correction and supporting outcome records. Explanations are model-generated interpretations, not privileged physical causality. Compact experience differs from an indefinitely growing transcript; its value depends on later behavioral benefit.

3.3 Experience-Conditioned Action Revision

The extracted experience is returned to the actor through the same context interface used for other auxiliary information. Let E_t denote the set of currently available extracted experiences. When a reusable experience e_t is produced,

$$E_{t+1} = \mathrm{Update}(E_t, e_t). \tag{5}$$

It can then participate in construction of the next actor context:

$$a_{t+1} \sim \pi_\theta\left(a \mid \mathrm{Compile}(o_{t+1}; \mathcal{S}_{t+1}), C\right), \qquad E_{t+1} \in \mathcal{S}_{t+1}. \tag{6}$$

Auxiliary sources can be independently enabled or removed. Raw history and extracted experience remain separate channels: the latter requires reflection and structured extraction before action revision. Nominal condition labels do not prove useful delivery, motivating information-flow auditing.

3.4 Recursive Self-Improvement

A single reflection followed by a corrected action corresponds to recovery. Our broader interest is whether this mechanism can operate **iteratively**: each new physical interaction can provide evidence that updates the context for the next decision, whose execution in turn produces additional evidence.

The complete recurrence is

$$\begin{aligned} a_t &\sim \pi_\theta\left(a \mid \mathrm{Compile}(o_t; \mathcal{S}_t), C\right), \\ o_{t+1} &\sim P(\cdot \mid o_t, a_t), \\ e_t &\leftarrow \mathrm{Reflect}(a_t, y_t, o_{t+1}), \\ \mathcal{S}_{t+1} &\leftarrow \mathrm{UpdateContext}(\mathcal{S}_t, e_t). \end{aligned} \tag{7}$$

Recursion occurs in the **behavior–experience–context loop**; actor weights, reflection and the action contract remain fixed. We do not assume monotonic improvement: reflection can misdiagnose failures, overfit corrections or retrieve irrelevant experience. Improvement must therefore be measured as appropriate action change and increased progress or completion under comparable physical conditions, rather than counted retries or textual reflections.

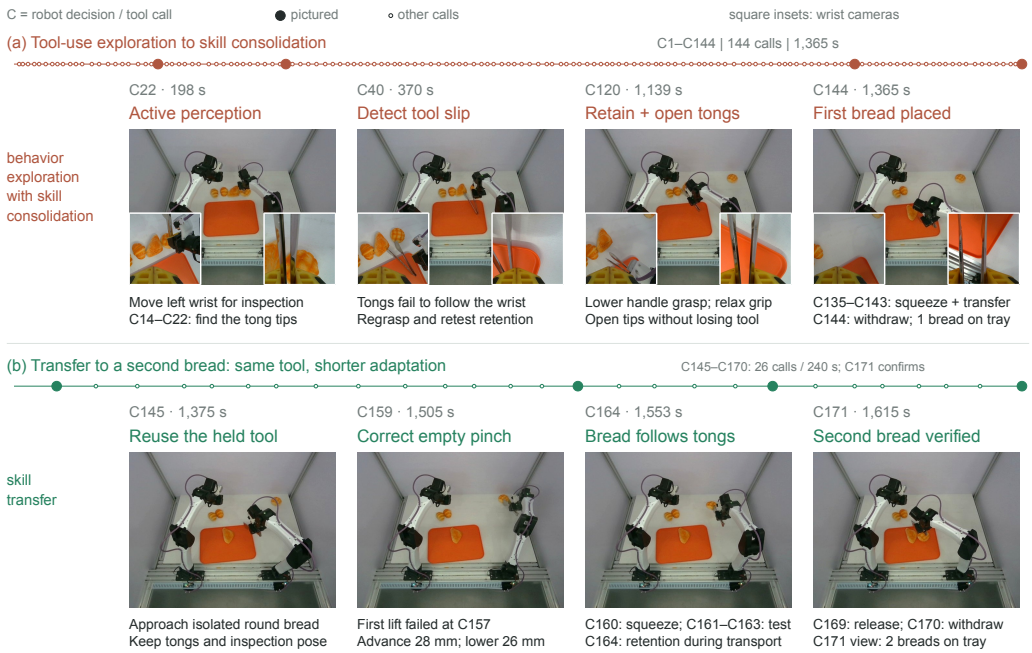

Figure 4: Physical exploration and within-session reuse in bread transfer. The first placement requires 144 calls and 1,365 s; the second requires 26 calls and 240 s, followed by verification at C171. Selected events show active perception, tool-slip recovery, tong actuation and target-specific correction. The second phase retains the tool and physical setup; the cost contrast is descriptive, not a causal learning estimate.

3.5 Context Management for Iterative Improvement

Experience is managed rather than appended indiscriminately. New lessons can be available immediately; when retrieval is enabled, the compiler selects a bounded number of relevant entries to control redundancy and contradiction. Consolidation and retrieval beyond immediate extraction are extensions, not prerequisites for the physical interaction–reflection–context loop.

4 Iterative Self-Improvement in Physical Manipulation

Figure 4 examines the mechanism in a bread-transfer task: the robot must grasp and actuate tongs, verify tool retention, pinch bread and transfer it onto a tray. The first transfer develops a workable tool-use strategy through exploration; the second reuses that structure while correcting a new failed pinch.

4.1 Learning Tool Use Through Physical Interaction

The first transfer requires 144 model decisions over 1,365 seconds. During this period, the actor does not immediately discover a successful trajectory. Instead, its behavior changes as physical interaction reveals previously unknown properties of the tool and its coupling with the robot. Early in the interaction, the geometry of the tong tips is difficult to determine from the available view. Between C14 and C22, the actor changes the wrist configuration to obtain a more informative observation. At C22, the wrist is positioned specifically for inspection of the tong tips. This illustrates that physical actions can serve an information-gathering role in addition to directly advancing task completion. Subsequent execution exposes a different problem. At C40, the actor observes that the tongs do not follow the wrist as expected, indicating that the current tool engagement is insufficient for reliable manipulation. It responds by regrasping the tool and testing whether the new grasp is retained. The relevant revision is therefore not simply another Cartesian target; the actor changes

how the tool itself is coupled to the robot. Further interaction reveals that acquiring the tongs is still insufficient: the robot must also open and close them without losing the tool. By C120, the actor uses a lower handle grasp and relaxes its grip sufficiently to open the tong tips while maintaining control of the tool. This provides a workable strategy for actuating the tongs rather than merely transporting them. With this interaction strategy established, the actor squeezes and transfers the first bread during C135–C143. At C144, the robot has withdrawn and one bread is visible on the tray. The sequence yields actionable structure: inspect uncertain geometry, verify tool retention, use handle engagement that permits actuation, and check capture before transport. Under the harness, such evidence can become structured experience rather than a full raw trajectory. These summaries remain tentative; the next phase tests whether useful interaction structure appears again.

4.2 Reuse on a Second Bread

After successfully placing the first bread, the actor proceeds to a second, isolated round bread. This phase requires 26 model calls from C145 to C170 and approximately 240 seconds, with C171 providing the final verification view. In contrast to the first phase, the actor does not repeat the extended process of discovering how to hold and actuate the tongs. At C145, it approaches the new target while retaining the tool and the inspection configuration already established during the previous interaction. Reuse does not eliminate execution errors. The first lift attempt at C157 fails because the bread is not successfully captured. The actor responds locally rather than restarting the full tool-discovery process. At C159, it advances approximately 28 mm and lowers approximately 26 mm before attempting another pinch. After squeezing at C160, the actor performs several test motions at C161–C163. At C164, the bread remains with the tongs during transport, providing evidence of successful retention. The actor subsequently releases the bread at C169 and withdraws at C170. The observation at C171 verifies that two breads are present on the tray. This second phase is important because the reused behavior is not a memorized sequence of Cartesian actions. The position of the second bread differs from that of the first, and the initial pinch still fails. What appears to be reused is a higher-level interaction strategy: retain control of the tool, approach the new target, test whether the pinch succeeds, and make a localized correction when the observed result is inconsistent with the intended outcome.

4.3 Reduced repeated exploration

The second phase shifts from broad tool discovery to localized grasp correction. Skill consolidation here means reusable interaction structure rather than a fixed motor trajectory: prior experience constrains subsequent reasoning while current observations resolve target-specific errors. Useful experience need not eliminate future failures.

4.4 Scope of the Evidence

The case connects capability and supplied-context responsiveness to guidance emerging from physical interaction. However, the phases start from different states: the second retains the tongs and inspection configuration, and uses a different bread and geometry. Its lower cost therefore cannot isolate extracted experience from retained physical state or continued history. Matched executions with and without the interaction-derived update, after scene and conversation resets, are required to quantify retained learning.

5 Related work

5.1 Pretrained foundation models for robotic manipulation

Robot-specialized action models and general-purpose reasoners provide different deployment interfaces. OpenVLA and $\pi_{0.5}$ learn visuomotor behavior from robot and broader training data (Kim et al., 2024; Physical Intelligence et al., 2025b); DreamZero jointly models future video and actions, while Cosmos Policy supports action generation and model-based planning (Ye et al., 2026; Kim et al., 2026). These are complementary sources of motor priors and prediction, not a historical sequence in which reasoning replaces control. SayCan, Code as Policies, VoxPoser and ReKep connect language to affordances, programs and geometric constraints (Ahn et al., 2022; Liang et al., 2022;

Huang et al., 2023; 2024). Concurrent Astra evaluations add evidence about deploying general-purpose models through robot interfaces (Cheng et al., 2026; Zhang et al., 2026a; Menon et al., 2026; Su et al., 2026). Our evaluation concerns the resulting model–context–interface system; task success alone does not identify intrinsic spatial reasoning or establish a universal model ranking.

5.2 In-context robot learning

Robot adaptation without target-task weight updates predates the models studied here. ICRT learns to condition actions on sensorimotor demonstration trajectories (Fu et al., 2024); RICL first trains a pretrained VLA for in-context adaptability, then retrieves demonstrations at deployment (Sridhar et al., 2025). RoboPrompt converts object poses and demonstration actions into textual context for off-the-shelf LLM action prediction (Yin et al., 2024). InCoRo combines an LLM controller with scene understanding and feedback for online adjustment (Zhu et al., 2024). These routes differ in pretraining, context representation and feedback access, but already establish that context can change robot behavior. DROC further distills language corrections into retrievable knowledge for new settings (Zha et al., 2023). We build on this literature rather than introduce a new in-context learning paradigm: the evaluation separates supplied-context use from the construction and later utility of self-derived corrections. History contrasts are preliminary context-use evidence, not sufficient evidence of retained learning.

5.3 Real-world robot adaptation and reinforcement learning

Real-world RL directly improves policies through interaction. SERL integrates sample-efficient off-policy learning with reward, reset and control infrastructure; HIL-SERL combines demonstrations and human interventions with online RL (Luo et al., 2024; 2025). Experience-based $\pi^*_{0.6}$ extends this route to pretrained VLAs (Physical Intelligence et al., 2025a). These systems address sample efficiency and execution robustness; freezing our model changes the adaptation interface, not the underlying need for valid feedback, exploration and physical evaluation. LMPC trains code-generating models to adapt more efficiently to human feedback (Liang et al., 2024). ReSYNC learns recovery skills, ASPIRE develops reusable programs, and ENPIRE supports heuristic and gradient-based physical improvement (Li et al., 2026; Lu et al., 2026; Xiao et al., 2026). CaP-X supplies benchmark and coding-agent improvement methods; SHAPER and Harness VLA study skill/harness evolution and memory-guided use of fixed primitives (Fu et al., 2026; Wang et al., 2026; Zhang et al., 2026b). Memory, contracts, retained corrections and validation are therefore established components, not standalone novelty claims for the proposed method.

6 Limitations and conclusion

Evidence is small, uneven and incompletely linked. Set C lacks continuous videos and online correction events; set E supplies illustrated operator feedback and wrist-camera views but not matched raw episode manifests. Selected examples do not supply a success-rate denominator. The bread phases share a session and retained physical setup, so their cost difference does not isolate learning. Fresh-reset tests of persistent updates remain unperformed. Optimizer novelty, guaranteed safe improvement and recursive learning-operator improvement are not established.

Physical experience can inform manipulation through an actionable context channel. The current evidence connects successful precision-demanding execution, behavior changes under supplied context, and exploration followed by local tool-use reuse. FreeMani organizes these ingredients into a testable context-update loop with fixed model weights. The next decisive result is whether the interaction-derived guidance, rather than an easier scene or continued history alone, improves new executions at an acceptable physical cost.

AI USE STATEMENT

AI tools assisted code inspection, source retrieval, auditing, drafting and visualization. Earlier-version generated conceptual artwork is retained in the source history, not presented as experimental imagery. Human authors remain responsible for verification.

ETHICS, REPRODUCIBILITY, AND AUTHOR DECLARATIONS

Supervised operation and emergency stopping remain necessary. Aggregate data and manuscript sources accompany the draft; raw-log release awaits approval. Author contributions, funding, conflicts, consent and institutional requirements await confirmation. No new robot runs or model-evaluation calls were made.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025. doi: 10.48550/arXiv.2507.19457. URL https://arxiv.org/abs/2507.19457.

Michael Ahn, Anthony Brohan, Noah Brown, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. doi: 10.48550/arXiv.2204.01691. URL https://arxiv.org/abs/2204.01691.

Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, Yong Mao, Ke Li, and Xing Sun. Training-free group relative policy optimization. *arXiv preprint arXiv:2510.08191*, 2025. doi: 10.48550/arXiv.2510.08191. URL https://arxiv.org/abs/2510.08191.

Dongzhou Cheng, Taoran Yi, Ye Fang, Xingwu Zhang, Fan Feng, Yixuan Li, Gengxiong Zhuang, Rongze Wang, Shuai Yang, Wei Song, Weizhi Xue, Minyan Wu, Jie Gui, Jiaqi Wang, and Tong Wu. In-context robot learning with VLM agents. *arXiv preprint arXiv:2609.19138*, 2026. doi: 10.48550/arXiv.2609.19138. URL https://arxiv.org/abs/2609.19138.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024. doi: 10.48550/arXiv.2408.15980. URL https://arxiv.org/abs/2408.15980.

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. doi: 10.48550/arXiv.2603.22435. URL https://arxiv.org/abs/2603.22435v2. Initial submission March 23, 2026; version 2 July 2, 2026.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. doi: 10.48550/arXiv.2307.05973. URL https://arxiv.org/abs/2307.05973.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. doi: 10.48550/arXiv.2409.01652. URL https://arxiv.org/abs/2409.01652.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. doi: 10.48550/arXiv.2406.09246. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026. doi: 10.48550/arXiv.2601.16163. URL https://arxiv.org/abs/2601.16163.

Bowen Li, Mayank Mishra, Y. Isabel Liu, Stone Tao, Nishanth Kumar, Alexander G. Gray, Ruwan Wickramarachchi, Jonathan Francis, Sebastian Scherer, and Tom Silver. Recover, discover, plan: Learning skills and concepts from robot failures. *arXiv preprint arXiv:2606.18328*, 2026. doi: 10.48550/arXiv.2606.18328. URL https://arxiv.org/abs/2606.18328.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022. doi: 10.48550/arXiv.2209.07753. URL https://arxiv.org/abs/2209.07753.

Jacky Liang, Fei Xia, Wenhao Yu, et al. Learning to learn faster from human feedback with language model predictive control. *arXiv preprint arXiv:2402.11450*, 2024. doi: 10.48550/arXiv.2402.11450. URL https://arxiv.org/abs/2402.11450.

Runyu Lu, Yubo Wu, Ethan Kou, Letian Fu, Wenli Xiao, Ajay Mandlekar, Yinzhen Xu, Guanya Shi, Ken Goldberg, Ang Chen, Mosharaf Chowdhury, Yuke Zhu, Linxi Fan, and Guanzhi Wang. ASPIRE: Agentic /skills discovery for robotics. *arXiv preprint arXiv:2607.00272*, 2026. doi: 10.48550/arXiv.2607.00272. URL https://arxiv.org/abs/2607.00272. Initial submission June 30, 2026.

Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. SERL: A software suite for sample-efficient robotic reinforcement learning. *arXiv preprint arXiv:2401.16013*, 2024. doi: 10.48550/arXiv.2401.16013. URL https://arxiv.org/abs/2401.16013.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025. doi: 10.1126/scirobotics.ads5033. URL https://doi.org/10.1126/scirobotics.ads5033.

Achu Menon, Sravanthi Machcha, Sabrina Zou, Tzu Kit Chan, and Jay Chooi. GPT-6 Astra on robotic manipulation. RoboCurve evaluation report, September 4, 2026. URL https://openai.robocurve.org/gpt-6-astra/. Accessed September 25, 2026.

OpenAI. GPT-6 Astra model. OpenAI API documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-6-astra. Accessed September 26, 2026.

Physical Intelligence, Ali Amin, Raichelle Aniceto, et al. $\pi^*_{0.6}$: a VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a. doi: 10.48550/arXiv.2511.14759. URL https://arxiv.org/abs/2511.14759.

Physical Intelligence, Kevin Black, Noah Brown, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b. doi: 10.48550/arXiv.2504.16054. URL https://arxiv.org/abs/2504.16054.

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. doi: 10.48550/arXiv.2508.02062. URL https://arxiv.org/abs/2508.02062.

Jiayi Su, Yixin Zheng, Mi Yan, Li Yi, Zhizheng Zhang, and He Wang. GPT 6 Astra as an embodied policy. Technical report and code, 2026. URL https://github.com/anonymous-report-421/GPT-as-Policy. Accessed September 25, 2026.

Peidong Wang, Zhiming Ma, Ying Chang, Xufang Luo, Yiqun Zhang, Zihan Wang, Xiaocui Yang, Shi Feng, Yuqing Yang, and Dongsheng Li. Self-evolving embodied agents via skill-harness evolution. *arXiv preprint arXiv:2608.11350*, 2026. doi: 10.48550/arXiv.2608.11350. URL https://arxiv.org/abs/2608.11350v2. Initial submission August 11, 2026; version 2 September 10, 2026.

Wenli Xiao, Jia Xie, Tonghe Zhang, Haotian Lin, Letian Fu, Haoru Xue, Jalen Lu, Yi Yang, Cunxi Dai, Zi Wang, Jimmy Wu, Guanzhi Wang, S. Shankar Sastry, Ken Goldberg, Linxi Fan, Yuke Zhu, and Guanya Shi. ENPIRE: Agentic robot policy self-improvement in the real world. *arXiv preprint arXiv:2606.19980*, 2026. doi: 10.48550/arXiv.2606.19980. URL https://arxiv.org/abs/2606.19980v2. Initial submission June 18, 2026; version 2 September 20, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. doi: 10.48550/arXiv.2602.15922. URL https://arxiv.org/abs/2602.15922.

Yida Yin, Zekai Wang, Yuvan Sharma, Dantong Niu, Trevor Darrell, and Roei Herzig. In-context learning enables robot action prediction in LLMs. *arXiv preprint arXiv:2410.12782*, 2024. doi: 10.48550/arXiv.2410.12782. URL https://arxiv.org/abs/2410.12782.

Lihan Zha, Yuchen Cui, Li-Heng Lin, Minae Kwon, Montserrat Gonzalez Arenas, Andy Zeng, Fei Xia, and Dorsa Sadigh. Distilling and retrieving generalizable knowledge for robot manipulation via language corrections. *arXiv preprint arXiv:2311.10678*, 2023. doi: 10.48550/arXiv.2311.10678. URL https://arxiv.org/abs/2311.10678.

Wenbo Zhang, Kaixuan Wang, Yutao Ouyang, Xiaoyu Huang, Liyang Li, Kailun Su, Weiyang Jin, Wenhao Chai, Haotian Liang, Zhiyang Dou, Yue Chen, and Tianxing Chen. An unexpected robot policy: Early evaluations of GPT-6 Astra on RoboDojo and beyond. *arXiv preprint arXiv:2609.24170*, 2026a. doi: 10.48550/arXiv.2609.24170. URL https://arxiv.org/abs/2609.24170.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Yi Nie, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Jiaji Rao, Zhengru Fang, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026b. doi: 10.48550/arXiv.2607.08448. URL https://arxiv.org/abs/2607.08448v5. Version 5, September 24, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. *arXiv preprint arXiv:2308.10144*, 2023. doi: 10.48550/arXiv.2308.10144. URL https://arxiv.org/abs/2308.10144.

Jiaqiang Ye Zhu, Carla Gomez Cano, David Vazquez Bermudez, and Michal Drozdzal. InCoRo: In-context learning for robotics control with feedback loops. *arXiv preprint arXiv:2402.05188*, 2024. doi: 10.48550/arXiv.2402.05188. URL https://arxiv.org/abs/2402.05188.