CRACKING MOBILE MANIPULATION: FROM FAILURE DIAGNOSIS TO AGENTIC INTERVENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent vision-language-action (VLA) models have shown strong performance on local robot manipulation, yet their success does not readily translate to long-horizon mobile manipulation. In our experiments, a pretrained subtask VLA achieves 80% success on atomic manipulation tasks but only 19% when these skills are composed into mobile manipulation. We argue that this gap stems from failures across the execution pipeline rather than from local manipulation capability alone. To uncover these bottlenecks, we systematically diagnose failures in long-horizon mobile manipulation and identify three dominant regimes: task planning, navigation and base positioning, and local manipulation. Based on this diagnosis, we develop a failure-guided agentic framework that augments a fixed subtask VLA with demonstration-based planning guidance, adaptive visual memory, accessibility-aware base adjustment, and local end-effector adjustment. On RoboCasa365, these interventions progressively improve task success from 19% to 47%, with the finalized system reaching 49% on held-out episode seeds, compared with 18% for the baseline. The same design improves performance from 14% to 37% on BEHAVIOR-1K and from 5/20 to 13/20 on real-robot tasks. These results suggest that reliable mobile manipulation can be advanced by diagnosing and intervening on the current bottleneck, rather than uniformly scaling the underlying policy.

1 INTRODUCTION

Recent vision-language-action (VLA) models have achieved strong performance on robot manipulation, particularly for tasks that can be completed within a local workspace (Zitkovich et al., 2023; Black et al., 2025b;a; Wen et al., 2025). Agentic systems further extend these capabilities to longer-horizon tasks through high-level planning, execution feedback, and tool use (Ichter et al., 2023; Zhang et al., 2026b; Fu et al., 2026). Yet mobile manipulation exposes a striking gap between solving individual subtasks and reliably composing them over time. Unlike tabletop manipulation, a mobile robot is required to move across workspaces while maintaining task progress, and every movement changes the viewpoint and base configuration from which the next manipulation is executed (Yenamandra et al., 2023; Fu et al., 2025; Liu et al., 2024; Wu et al., 2025). As a result, strong local manipulation skills do not necessarily translate into reliable long-horizon mobile manipulation.

The gap is surprisingly large. As shown in Figure 1, a pretrained subtask VLA achieves **80%** success on atomic manipulation tasks, yet the corresponding agentic baseline succeeds on only **19%** of long-horizon mobile manipulation tasks. This discrepancy suggests that the central challenge is not simply whether the robot knows how to manipulate, but whether it can reliably carry a manipulation skill through the sequence of decisions and physical states required by mobile manipulation tasks. A long-horizon task can fail before manipulation even begins: the planner may select the wrong target or lose track of task progress; navigation may reach the correct workspace but leave the robot in an unusable base pose; and even from a seemingly appropriate configuration, small errors in localization or end-effector alignment can derail the final interaction. In this sense, what reliable mobile manipulation requires is not simply making a local skill stronger, but identifying where the execution pipeline breaks down.

This leads to a fundamental question: ***Where does mobile manipulation actually fail over long-horizon tasks, and how can an agent intervene at the right bottleneck?*** We answer this question by cracking the failure structure of mobile manipulation. Starting from a pretrained VLA that already

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.

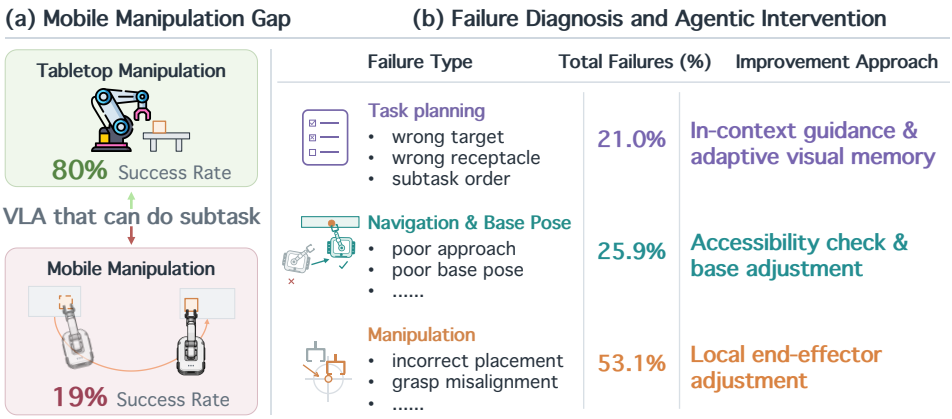

Figure 1: **From the mobile manipulation gap to failure-guided intervention.** (a) Strong subtask-level manipulation does not directly translate to reliable long-horizon mobile manipulation. (b) Failure analysis identifies the dominant failures as task planning, navigation and base positioning, and manipulation. These observations motivate our agentic intervention, including: in-context guidance from video demonstrations, adaptive visual memory, accessibility checking and base adjustment, and end-effector adjustment.

performs individual tabletop subtasks well, we systematically analyze its failures when these subtasks are composed with navigation. Our analysis reveals three dominant failure regimes: task planning, navigation and base positioning, and local manipulation. More importantly, these failures form a natural progression through the execution pipeline. Planning errors prevent the robot from pursuing the correct subtask; navigation errors prevent it from reaching an executable configuration; and manipulation errors remain after the robot has successfully reached the intended workspace. This diagnosis motivates a simple but powerful principle: intervene where the current system breaks, rather than uniformly increasing the capability of the underlying policy.

Guided by this principle, we develop a failure-guided agentic framework that progressively adds targeted interventions around a fixed subtask VLA. For planning, in-context guidance from video demonstrations provides explicit task structure, while adaptive visual memory retains the observations needed to track what has already been accomplished and what remains. For navigation, an accessibility check evaluates whether the current robot configuration is suitable for the upcoming manipulation, followed by base adjustment when the approach is geometrically inadequate. Finally, for local execution, end-effector adjustment refines the interaction pose to compensate for residual spatial misalignment. These components are not independent add-ons: each is directly motivated by a diagnosed failure mode, transforming failure analysis into targeted agentic intervention.

We evaluate the resulting system on all mobile manipulation composite tasks in RoboCasa365 (Nasiriany et al., 2026). Despite its 80% atomic-task success, the agentic baseline achieves only 19% task success in the long-horizon mobile setting. Progressively introducing the proposed interventions raises success to 47%, while the finalized system reaches 49% on held-out episode seeds, compared with 18% for the baseline. The same design also improves performance beyond RoboCasa365, improving task success from 14% to 37% on five BEHAVIOR-1K tasks (Li et al., 2023) and from 5/20 to 13/20 on two real-robot tasks. Taken together, these results suggest a different way to approach long-horizon mobile manipulation. **Rather than treating the robot as a monolithic policy whose capability must be scaled uniformly, we view it as an agentic system whose reliability can be improved by diagnosing bottleneck and intervening it precisely.**

Our contributions are summarized as follows:

- **A systematic failure diagnosis of mobile manipulation.** We analyze the gap between strong subtask-level manipulation and long-horizon mobile manipulation, identifying three dominant failure regimes: task planning, navigation and base positioning, and local manipulation. The analysis turns overall success gap into a concrete, stage-wise failure structure.
- **A failure-guided agentic intervention framework.** Based on the diagnosed bottlenecks, we introduce targeted interventions around a fixed subtask VLA, including demonstration-

based planning guidance, adaptive visual memory, accessibility-aware base adjustment, and local end-effector adjustment.

- **Agentic intervention consistently improves the performance of mobile manipulation.** The proposed interventions improve RoboCasa365 task success from 19.0% to 47.0%. Our experiments further demonstrate consistent gains on held-out RoboCasa365 episodes, BEHAVIOR-1K tasks, and a real dual-arm mobile manipulator.

2 Related Work

2.1 Mobile Manipulation Systems and Policies

Mobile manipulation requires coordinating base motion and manipulation over long-horizon tasks. Existing approaches span modular systems that separate perception, navigation, and manipulation (Yenamandra et al., 2023; Liu et al., 2024), and learned policies that couple mobility and manipulation through teleoperated demonstrations, scene representations, whole-body control, or trajectory models (Fu et al., 2025; Uppal et al., 2024; Qiu et al., 2025; Yan et al., 2025; Sundaresan et al., 2025).

With larger robot datasets and pretrained models (Ye et al., 2026), generalist policies have scaled toward multi-task and cross-embodiment control (Brohan et al., 2023; Open X-Embodiment Collaboration, 2024; Ghosh et al., 2024; Liu et al., 2025; Li et al., 2026b). Vision-language-action models further extend this paradigm to general robot control (Zitkovich et al., 2023; Black et al., 2025b;a; Wen et al., 2025; Gemini Robotics Team et al., 2025; NVIDIA et al., 2025; NVIDIA, 2025a;b; Zheng et al., 2026; Wu et al., 2026; Yu et al., 2026; Xiaomi Robotics Team et al., 2026; Yuan et al., 2026a; Kim et al., 2026a), while recent work targets mobile manipulation through fixed-base VLA transfer, whole-body action prediction, spatially enriched observations, and action-conditioned world models (Wu et al., 2025; Tu et al., 2026; Yang et al., 2026a; Kim et al., 2026b; Yuan et al., 2026b; Chen et al., 2026; Ma et al., 2026). System-level approaches additionally incorporate structured planning, manipulation-aware base positioning, and recovery (Li et al., 2026a; Lim et al., 2026). Our focus is complementary: we study how failures are distributed across task planning, navigation and base positioning, and local manipulation in an agentic system, and how the dominant bottleneck shifts as targeted interventions are introduced.

2.2 Robot Failure Diagnosis and Recovery

Task success rate alone provides limited information about why a robot fails. Recent benchmarks evaluate policies along finer-grained capability and failure dimensions (Gao et al., 2026; Yu & Qiu, 2026), while diagnosis methods use multimodal execution traces or collected failure data to identify failure causes and generate corrective feedback (Liu et al., 2023; Duan et al., 2025; Lin et al., 2025; Ye et al., 2025; Zeng et al., 2026). Other methods integrate failure handling directly into execution through VLM supervisors, failure detectors, recovery policies, or world-model-based verification (Chen et al., 2025; Yang et al., 2026b; Zhao et al., 2026; Luo et al., 2026; Seligmann et al., 2026; Liu et al., 2026; Zhang et al., 2026a). Rather than focusing on failure detection or recovery itself, we analyze where failures arise in an agentic mobile manipulation system, use this diagnosis to guide targeted interventions, and track how the remaining failure distribution shifts as the system improves.

2.3 Language-Model Robot Agents, Demonstrations, and Memory

Language and vision-language models have been widely used for robot task planning, semantic grounding, and closed-loop execution (Ichter et al., 2023; Huang et al., 2023c; Wang et al., 2025; Driess et al., 2023; Huang et al., 2023b; Rana et al., 2023; Chen et al., 2024; Sun et al., 2025). Programmatic and coding-agent-based systems further compose perception, planning, and control tools through executable programs, iterative feedback, debugging, and skill refinement (Liang et al., 2023; Singh et al., 2023; Huang et al., 2023a; Fu et al., 2026; Lu et al., 2026; Xiao et al., 2026; Zhang et al., 2026b), with recent work also exploring tighter integration between VLAs, external tools, and specialized robot policies (Lei et al., 2026; Shi et al., 2026).

Demonstrations can provide task specifications or planning guidance for long-horizon manipulation (Wang et al., 2023; Jain et al., 2024; Wake et al., 2024), while memory and reusable skills help agents retain task progress and reuse prior experience (Yoneda et al., 2024; Zhang et al., 2026b; Shi et al., 2026). In our failure-guided agentic framework, the skill library provides demonstration-based planning guidance, while adaptive visual memory gives the planner on-demand access to relevant observations from the current episode. Both serve as targeted interventions around a fixed subtask VLA rather than directly modifying the low-level policy.

3 Diagnosing Failures in Agentic Mobile Manipulation

We analyze where the agentic baseline fails on long-horizon mobile manipulation. We first describe the evaluation tasks and baseline, then define the failure attribution procedure and summarize the main failure patterns.

3.1 Diagnostic Setting

We study all 20 composite tasks of RoboCasa365 that require mobile manipulation. These tasks contain 2–15 subtasks and require the robot to move between workspaces while performing object transport, pick-and-place, and articulated-object interaction. Additional task details are provided in Appendix B.

Our baseline follows a hierarchical agentic design adapted from Harness VLA (Zhang et al., 2026b). A high-level planner selects subtask instructions or tools, while a shared subtask VLA executes both navigation and manipulation. The subtask VLA is trained on RoboCasa365 atomic task demonstrations and subtask-level segments from the training split, and achieves 80% success on atomic-task evaluation of RoboCasa365. However, the complete agentic system reaches only 19% success rate on the long-horizon mobile manipulation tasks. This gap motivates our central question: *where does a system with capable local skills fail when these skills must be coordinated over a long horizon?* Implementation and training details are provided in Appendix C.

3.2 Failure Attribution

We evaluate the baseline on the 20 mobile manipulation tasks with five rollouts per task, giving 100 episodes in total. The baseline succeeds in 19 episodes, leaving 81 failed episodes for failure analysis. For an episode trajectory, $\tau = (o_0, a_0, \ldots, o_T)$, we define t^* as the earliest step at which an error occurs that affects the final task outcome and is not later recovered. We assign the failure at t^* as the *primary failure* of the episode. This definition identifies the first error that remains unrecovered and affects the final task outcome, rather than counting later errors caused by the same failure.

For offline diagnosis, we inspect the rollout video together with the agent interaction history, robot states, and simulator object states. Simulator states are used only for diagnosis and are not available to the online agent. All failure labels are reviewed by human annotators. On a random subset of 30 failed episodes, three reviewers independently agree on 26 cases, with Fleiss' $\kappa = 0.86$. Details of the diagnosis procedure, prompting, and annotation consistency are provided in Appendix D.

3.3 Failure Analysis

Finding 1: Long-horizon task planning fails when task progress is not tracked reliably. In many task-planning failures, the required low-level skills are available, but the planner applies them to the wrong target or at the wrong stage of the task. In long-horizon execution, an early mistake can change the scene and make later actions ineffective. Reliable planning therefore requires tracking what has happened and what remains to be done.

Finding 2: reaching the correct area does not guarantee a suitable base pose for manipulation. Many navigation failures occur after the robot reaches the correct part of the scene but stops at a base pose that is poor for the next manipulation. Base pose determines the arm workspace so that navigation should place the robot at a pose suitable for next interaction, rather than simply near the target.

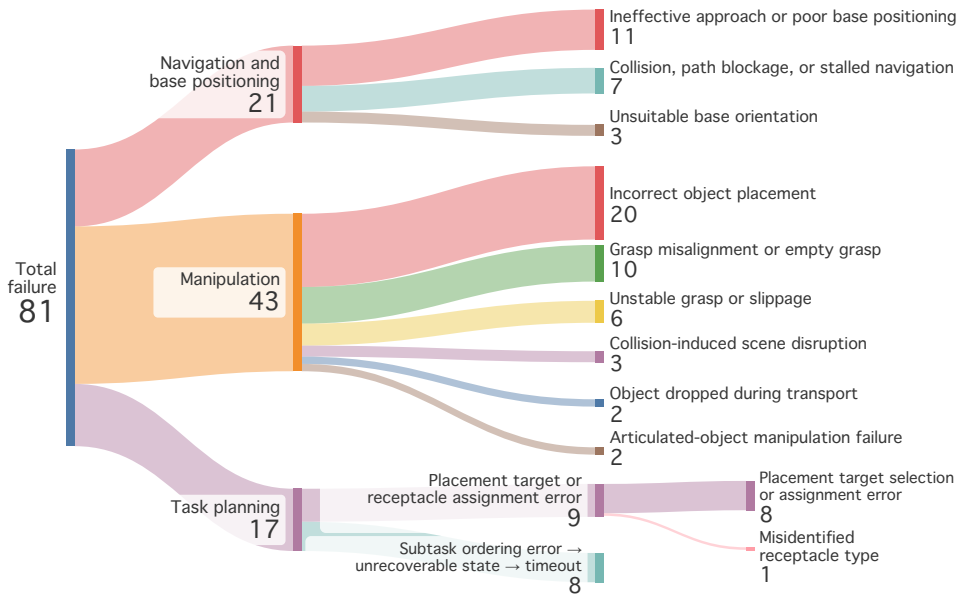

Figure 2: **Failure attribution of the agentic baseline.** Among the 81 failed episodes, primary failures are attributed to task planning (21.0%), navigation and base positioning (25.9%), and manipulation (53.1%).

Finding 3: Manipulation failures often remain after the correct task planning and workspace are reached. Errors in target localization or end-effector alignment can still cause a grasp or placement to fail. This motivates a local adjustment before retrying the interaction.

Taken together, these findings point to three weak links in long-horizon mobile manipulation: tracking task progress, handing off from navigation to manipulation, and turning a correct semantic decision into a precise target interaction.

4 Failure-Guided Agentic Intervention

Guided by the failure analysis in Section 3, we keep the subtask VLA fixed and add targeted interventions at the diagnosed bottlenecks. These include demonstration-based planning guidance from the skill library, adaptive visual memory, accessibility-aware base adjustment, and end-effector adjustment. Figure 3 summarizes the resulting failure-guided agentic framework.

4.1 Skill Library for Demonstration-based in context Guidance

Task planning failures often involve incorrect target selection or subtask order. We use demonstration videos to provide task-level guidance to the planner. For each evaluation task, a VLM matches the task instruction to task annotations in the benchmark training split and retrieves the most similar demonstration. Gemini 3.5 Flash summarizes the demonstration into a compact skill card containing the task goal, an ordered subtask plan, representative keyframes, and completion criteria. The planner uses the text fields and representative keyframes in this card together with the current observation and execution history when selecting the next step.

4.2 Adaptive Visual Memory

The planner observes the textual history of previous tool calls and execution results, but this history may not preserve visual changes or other information needed to track progress in a long-horizon task. We therefore provide on-demand access to visual observations collected during the current episode. Given a natural-language query from the planner, a separate VLM (Qwen3.8-27B) ranks the stored visual observations by their relevance to the query and retrieves the 16 most relevant frames. These frames are returned to the planner together with a short summary of the corresponding trajectory history. The planner decides when memory is needed and formulates a query based on its current task state. Retrieval uses only observations collected before the current agent step and has no access

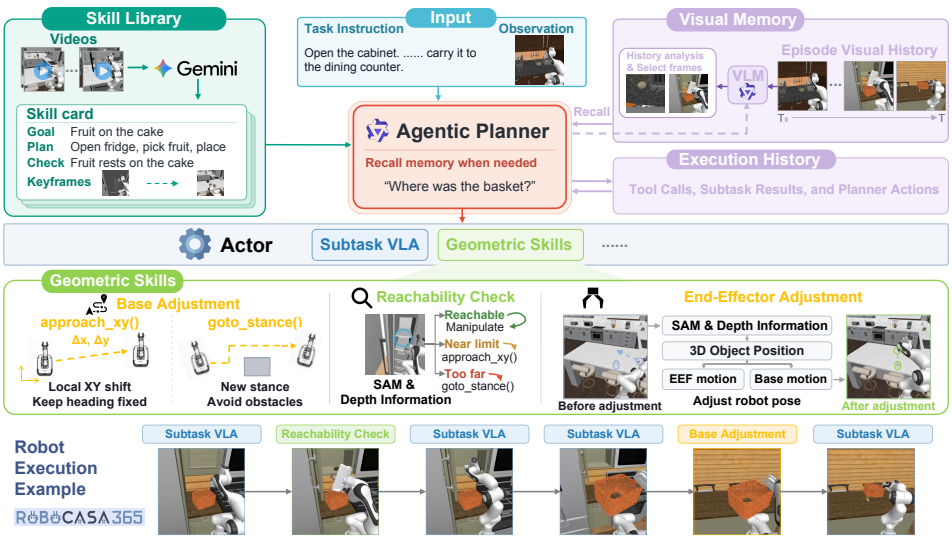

Figure 3: **Overview of the failure-guided agentic system.** The high-level planner uses skill library for task planning guidance, adaptive visual memory for relevant observation retrieval, and geometric skills for accessibility checking, base adjustment, and end-effector adjustment.

to future observations or simulator states. For example, when the planner asks "Where was the basket last seen?", the memory module retrieves the relevant observations and summarizes the corresponding history.

4.3 Geometric Skills

The remaining two interventions address navigation and manipulation failures. Base adjustment changes the robot pose when the current stance is poor for manipulation, while end-effector adjustment applies a local correction when grasping or placement is misaligned.

Accessibility check and base adjustment. Before transitioning from navigation to manipulation, the agent calls `check_stance` to estimate whether the current base pose is suitable for interacting with the target. The current subtask specifies the target object or interaction region, whose name is used as a text query for Grounding DINO. SAM then refines the resulting target mask. Valid depth pixels within the mask are back-projected using the calibrated camera parameters to estimate the target position in the robot coordinate frame. We classify the current stance as reachable, marginal, or out of reach based on the target distance. The thresholds are estimated from successful training trajectories and fixed before evaluation. This provides a coarse check of whether the base is at a suitable distance for interaction, rather than a full manipulation feasibility test that accounts for inverse kinematics, joint limits, or full-body collisions. For a marginal stance, the agent calls `approach_xy` to apply a small local base correction while keeping the heading fixed. For an out-of-reach stance, `goto_stance` localizes the target from the current observations, selects a new base stance within the manipulation range, and plans a path to that stance using A^* on a 2D occupancy grid constructed from the current depth observations. Observed obstacles are inflated by 0.1 m before planning, and the base follows the resulting waypoints using a PD controller. The distance thresholds and main geometric-skill parameters are provided in Appendix H.

End-effector adjustment. For grasping and placement, the current subtask specifies the target object or placement region. We use the same RGB-D grounding pipeline to obtain the target mask. Valid depth pixels within the mask are back-projected into 3D, and their median position is used as the target position to reduce sensitivity to depth noise. A local surface normal is estimated from the same 3D points and used to define the end-effector approach direction. The agent then applies a bounded adjustment toward an offset pose along this direction, with translation limited to 0.3 m per call.

5 Experiments and Results

We evaluate the framework on long-horizon mobile manipulation and analyze how the remaining failures change as components are added. We first study progressive integration and failure changes on the development episodes of RoboCasa365. After fixing the system design, we evaluate on held-out episode seeds. We then test the same agentic design on BEHAVIOR-1K (Appendix G) and on a real mobile manipulator (Appendix J). We also compare with end-to-end and agentic baselines and examine key design choices through ablations.

5.1 Evaluation Protocol

We evaluate all methods on the tasks setting described in Section 3, using five rollouts per task. The development evaluation uses the same episode seeds and simulator initial states across configurations. After fixing the system design, prompts, and tool parameters, we evaluate on a second set of five held-out episode seeds per task, using the same task definitions but different scene and object initial states. Unless otherwise stated, all agentic methods use Kimi K3 (AI, 2026) and the same fixed subtask VLA. Task success rate is measured using the official RoboCasa365 evaluator. Paired statistical analyses are provided in Appendix I. We additionally evaluate the framework on a real mobile manipulator; the setup and results are reported in Appendix J.

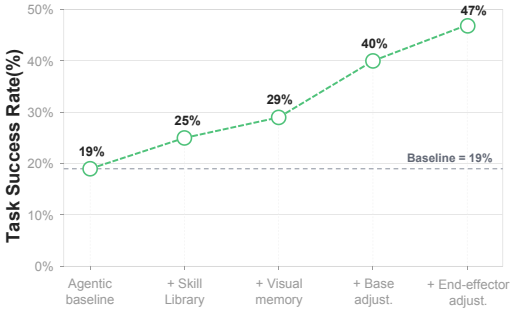

Figure 4: **Progressive integration of failure-guided interventions.** Each point adds one intervention group to the previous configuration.

5.2 Progressive Integration

Starting from the agentic baseline, we progressively add skill library, adaptive visual memory, base adjustment, and end-effector adjustment, while keeping the subtask VLA fixed. Progressive integration improves task success rate from 19.0% to 47.0%. As shown in Figure 4, base adjustment provides the largest incremental gain in this sequence, increasing success from 29.0% to 40.0%. Skill library and adaptive visual memory reduce primary task-planning failures, while base adjustment reduces navigation and base-positioning failures. The paired analysis in Appendix D.2 shows that six baseline planning failures and six navigation/base-positioning failures become manipulation failures in the full system. These episodes reach later stages after earlier errors are addressed, leaving manipulation as the largest remaining failure category.

A finer-grained analysis shows a similar pattern. Several failure modes directly targeted by the interventions are reduced, including placement-target or receptacle assignment errors (9 to 0), ineffective approaches or poor base positioning (11 to 1), and grasp misalignment or empty grasps (10 to 3). The remaining failures are dominated by incorrect object placement, object drops during transport, and collision-induced scene disruption, which are not fully addressed by the current geometric corrections. The complete fine-grained breakdown is provided in Appendix D.3.

5.3 Main Comparison and Held-Out Evaluation

Table 1 compares the full system with representative end-to-end VLA policies and agentic system. All methods are evaluated on the same episode seeds. Agentic baselines additionally use the same planner budget and retry budget, while Pigey and our Harness VLA adaptation use the same subtask VLA checkpoint as our method. For the end-to-end baselines, we use the released checkpoints and corresponding implementations linked from the official RoboCasa365 repository (Appendix C).

Compared with the Harness VLA baseline, our full system increases task success rate from 19.0% to 47.0% on the development episodes and from 18.0% to 49.0% on the held-out episodes. The held-out improvement remains significant under paired analysis ($p < 0.001$), suggesting that the gain is not limited to the episode seeds used for system development. The full system also achieves performance

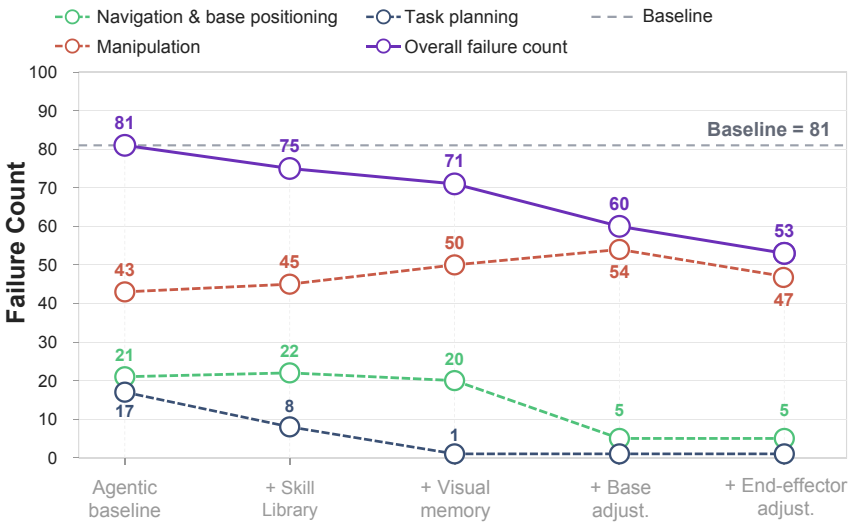

Figure 5: **Primary failure distribution during progressive integration.** Counts are reported over the same 100 matched episodes.

comparable to the strongest end-to-end VLA baseline, Xiaomi Robotics 1. On five BEHAVIOR-1K tasks with matched initial states, it improves success rate from 14.0% to 37.0% over the Harness VLA baseline. The low-level policies are trained on BEHAVIOR-1K training data for this evaluation. The further details are provided in Appendix G.

Table 1: **Performance of mobile manipulation tasks on the RoboCasa365 benchmark.** We evaluate the 20 of 32 composite tasks that require mobile manipulation.

Type	Method	Development (%)	Held-out (%)
VLA	π_0 (Black et al., 2025b)	2.0	2.0
	$\pi_{0.5}$ (Black et al., 2025a)	3.0	4.0
	GR00T N1.6 (NVIDIA, 2025b)	7.0	8.0
	GR00T N1.5 (NVIDIA, 2025a)	8.0	9.0
	RLDX-1 (Kim et al., 2026a)	15.0	14.0
	Xiaomi Robotics 1 (Xiaomi Robotics Team et al., 2026)	44.0	45.0
Agentic	CaP-X (Kimi K3) (Fu et al., 2026)	10.0	11.0
	Pigey (Kimi K3) (Galanti et al., 2026)	13.0	12.0
	Harness VLA (Kimi K3) (Zhang et al., 2026b)	19.0	18.0
	Ours (Kimi K3)	**47.0**	**49.0**

5.4 Ablation Study

Because the progressive experiment is cumulative, its incremental gains depend on the order of integration. Therefore, Appendix F reports leave-one-out evaluations from the full system. Here, we focus on the main design choices within each capability. All ablations use the same 100 matched episodes. Paired comparisons for these design choices are reported in Appendix I.4.

Skills Library. We ablate both the demonstration source and the visual content provided to the planner. HIW-500 is a real-world humanoid dataset of teleoperated household demonstrations, while our in-domain demonstrations come from the RoboCasa365 training split. With the same text skill card and demonstration keyframes, using in-domain demonstrations increases the observed success rate from 40.0% to 47.0%. When using the same in-domain demonstrations, removing the keyframes and retaining only the text skill card gives 43.0%. These results suggest that both domain alignment and visual demonstration context can affect the usefulness of demonstration guidance.

Adaptive visual memory. We compare always-provided visual history with planner-triggered memory access. In the former, 8, 16, or 32 uniformly sampled history frames are provided at every planner turn. In our adaptive memory, the planner decides when to query the history, and each query

Table 2: **Design ablations on 20 RoboCasa365 mobile manipulation tasks.** Each group varies one design choice while keeping the remaining components of the full system unchanged.

Component	Variant	Success Rate (%)
Reference	**Full system**	**47.0**
Skill Library	Text-only skill card (VLA training data)	43.0
	Text skill card + Keyframes (HIW-500)	40.0
	Text skill card + keyframes (VLA training data)	**47.0**
Adaptive Visual Memory	Uniform history every turn (8 frames)	37.0
	Uniform history every turn (16 frames)	40.0
	Uniform history every turn (32 frames)	35.0
	Planner-triggered memory query (8 frames)	42.0
	Planner-triggered memory query (16 frames)	**47.0**
	Planner-triggered memory query (32 frames)	45.0
Base Adjustment	w/o accessibility check (`check_stance`)	36.0
	w/o local base positioning adjustment (`approach_xy`)	42.0
	w/o stance replanning (`goto_stance`)	39.0
End-effector Adjustment	target location by agentic planner	43.0
	target location by geometric skills	**47.0**

returns the specified number of frames relevant to the planner's request. Planner-triggered memory achieves a higher observed success rate at each frame budget, while increasing the number of frames does not consistently improve performance. This suggests that selectively accessing task-relevant history is more useful than continuously providing more visual context.

Base adjustment. Removing the accessibility check causes the largest drop among the base-adjustment variants. Removing stance replanning also has a larger effect than removing the local base positioning adjustment. This pattern is consistent with our failure analysis: explicitly checking whether the current pose is suitable for manipulation, and replanning when it is not, are particularly important at the navigation–manipulation transition.

End-effector adjustment. We compare two ways of specifying the 3D target for local end-effector adjustment. Letting the planner specify the target directly achieves 43.0% success, compared with 47.0% when the target is estimated from RGB-D geometry. This modest numerical difference suggests that explicit geometric grounding can reduce the burden on the high-level planner for precise local target specification.

6 Conclusion

We presented a failure-guided approach to long-horizon mobile manipulation that moves beyond simply improving the underlying manipulation policy. Starting from a pretrained VLA with strong subtask-level performance, we systematically identified where mobile manipulation breaks down and found three dominant failure regimes: task planning, navigation and base positioning, and local manipulation. We then translated these observations into targeted agentic interventions, including demonstration-based planning guidance, adaptive visual memory, accessibility-aware base adjustment, and local end-effector adjustment. Across RoboCasa365, BEHAVIOR-1K, and real-robot experiments, these interventions substantially improve long-horizon task success while leaving the underlying subtask VLA fixed. This provides a useful perspective on agentic robot learning: progress can come not only from making the policy stronger, but from understanding what currently limits it and intervening at that bottleneck.

More broadly, we hope this work encourages failure diagnosis to become a first-class tool for designing embodied agents. As robots are deployed in increasingly long-horizon, multi-stage environments, their dominant failure modes may evolve throughout execution and as capabilities improve. Identifying and addressing these changing bottlenecks could provide a systematic path toward more reliable and capable mobile manipulation.

AI USE STATEMENT

Generative AI tools were used to assist with literature collection, partial manuscript drafting and language polishing, as well as the implementation and debugging of parts of the experimental code. The research ideas, methodology, experimental design, analysis, and conclusions were developed and verified by the authors. All experimental results reported in this paper were obtained from the authors' actual experiments.

REPRODUCIBILITY STATEMENT

The appendix reports the evaluation task set, subtask VLA training setup, failure attribution procedure, and additional agent-side skill details. Code, prompts, configuration files, and evaluation episode lists will be released to support reproduction of the reported experiments.

REFERENCES

Moonshot AI. Kimi k3: Open frontier intelligence. *Technical release announcement, July*, 2026.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://doi.org/10.15607/RSS.2025.XXI.010.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H. Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://doi.org/10.15607/RSS.2023.XIX.025.

Hongyi Chen, Yunchao Yao, Ruixuan Liu, Changliu Liu, and Jeffrey Ichnowski. Robot failure recovery using vision-language models with optimized prompts. In *2025 American Control Conference (ACC)*, pp. 1983–1988. IEEE, 2025. doi: 10.23919/ACC63710.2025.11107751. URL https://doi.org/10.23919/ACC63710.2025.11107751.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026. URL https://arxiv.org/abs/2607.00678.

Yongchao Chen, Jacob Arkin, Charles Dawson, Yang Zhang, Nicholas Roy, and Chuchu Fan. AutoTAMP: Autoregressive task and motion planning with llms as translators and checkers. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6695–6702. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611163. URL https://doi.org/10.1109/ICRA57147.2024.10611163.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A Vision-Language-Model for Detecting and Reasoning Over Failures in Robotic Manipulation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/70a06501001e1820fd1eb9ee821302d2-Abstract-Conference.html.

Letian Fu, Justin Yu, Karim El-Refai, Ethan Kou, Haoru Xue, Huang Huang, Wenli Xiao, Guanzhi Wang, Dantong Niu, Fei-Fei Li, et al. Cap-x: A framework for benchmarking and improving coding agents for robot manipulation. *arXiv preprint arXiv:2603.22435*, 2026.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning Bimanual Mobile Manipulation using Low-Cost Whole-Body Teleoperation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4066–4083. PMLR, 2025. URL https://proceedings.mlr.press/v270/fu25b.html.

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.

Ning Gao, Jinliang Zheng, Xing Gao, Haoxiang Ma, Hanqing Wang, Yukai Wang, Jiantong Chen, Zanxin Chen, Shujie Zhang, Mingda Jia, Xuekun Jiang, Zihou Zhu, Xinyu Li, Shuai Wang, Hao Li, Wenzhe Cai, Yuqiang Yang, Xudong Xu, Zhaoyang Lyu, Yao Mu, Tai Wang, Jiangmiao Pang, Jia Zeng, Weinan Zhang, and Chunhua Shen. EBench: Elemental Diagnosis of Generalist Mobile Manipulation Policies. *arXiv preprint arXiv:2606.18239*, 2026. URL https://arxiv.org/abs/2606.18239.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. URL https://arxiv.org/abs/2503.20020.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090. URL https://doi.org/10.15607/RSS.2024.XX.090.

Siyuan Huang, Zhengkai Jiang, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Instruct2act: Mapping multi-modality instructions to robotic actions with large language model. *arXiv preprint arXiv:2305.11176*, 2023a. URL https://arxiv.org/abs/2305.11176.

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. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562. PMLR, 2023b. URL https://proceedings.mlr.press/v229/huang23b.html.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782. PMLR, 2023c. URL https://proceedings.mlr.press/v205/huang23c.html.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T. Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

Vidhi Jain, Maria Attarian, Nikhil J. Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R. Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2Robot: End-to-end Video-conditioned Policy Learning with Cross-Attention Transformers. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.052. URL https://doi.org/10.15607/RSS.2024.XX.052.

Dongyoung Kim, Huiwon Jang, Myungkyu Koo, Suhyeok Jang, Taeyoung Kim, Beomjun Kim, Byungjun Yoon, Changsung Jang, Daewon Choi, Dongsu Han, et al. RLDX-1 technical report. *arXiv preprint arXiv:2605.03269*, 2026a. URL https://arxiv.org/abs/2605.03269.

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. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

Zixing Lei, Changxing Liu, Yichen Xiong, Minhao Xiong, Yuanzhuo Ding, Zhipeng Zhang, Weixin Li, and Siheng Chen. Towards Long-horizon Embodied Agents with Tool-Aligned Vision-Language-Action Models. *arXiv preprint arXiv:2605.13119*, 2026. URL https://arxiv.org/abs/2605.13119.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pp. 80–93. PMLR, 2023.

Chenxu Li, Zixuan Chen, Yetao Li, Jiapeng Xu, Hongyu Ding, Jieqi Shi, Jing Huo, and Yang Gao. MoMaStage: Skill-State Graph Guided Planning and Closed-Loop Execution for Long-Horizon Indoor Mobile Manipulation. *arXiv preprint arXiv:2603.08383*, 2026a. URL https://arxiv.org/abs/2603.08383.

Xinghang Li, Peiyan Li, Long Qian, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Xinlong Wang, Di Guo, Tao Kong, Hanbo Zhang, and Huaping Liu. What matters in building vision–language–action models for generalist robots. *Nature Machine Intelligence*, 8: 158–172, 2026b. doi: 10.1038/s42256-025-01168-7. URL https://doi.org/10.1038/s42256-025-01168-7.

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. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10160591. URL https://doi.org/10.1109/ICRA48891.2023.10160591.

Jia Syuen Lim, Zhizhen Zhang, Peter Bohm, Brendan Tidd, Zi Huang, and Yadan Luo. AnchorVLA: Anchored Diffusion for Efficient End-to-End Mobile Manipulation. *arXiv preprint arXiv:2604.01567*, 2026. URL https://arxiv.org/abs/2604.01567.

Zijun Lin, Jiafei Duan, Haoquan Fang, Dieter Fox, Ranjay Krishna, Cheston Tan, and Bihan Wen. Failsafe: Reasoning and recovery from failures in vision-language-action models. *arXiv preprint arXiv:2510.01642*, 2025.

Peiqi Liu, Yaswanth Orru, Jay Vakil, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Demonstrating OK-Robot: What Really Matters in Integrating Open-Knowledge Models for Robotics. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.091. URL https://doi.org/10.15607/RSS.2024.XX.091.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.07864.

Yushan Liu, Peibo Sun, Xintao Chao, Zhenyang Yang, Yifan Xie, Lingfeng Zhang, Shoujie Li, Chenyu Tang, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. CheckVLA: Execution-time verification with action-conditioned world model for long-horizon mobile manipulation. *arXiv preprint arXiv:2607.26789*, 2026. URL https://arxiv.org/abs/2607.26789.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3468–3484. PMLR, 2023. URL https://proceedings.mlr.press/v229/liu23g.html.

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. URL https://arxiv.org/abs/2607.00272.

Haojie Luo, Xizhou Bu, Hongbo Wang, Shijie Guo, Junxiu Liu, and Wei Li. Failure detection with zero-shot error correction in robotic manipulation. *IEEE Robotics and Automation Letters*, 11 (5):6256–6263, May 2026. doi: 10.1109/LRA.2026.3678124. URL https://doi.org/10.1109/LRA.2026.3678124.

Siyuan Ma, Boshi Zhang, Yutian Zhang, Qinglian Wu, Jiaqi Zhai, Dong Wei, and Qiaojun Yu. DECOWAM: Decoupled whole-body world-action model for legged mobile manipulation. *arXiv preprint arXiv:2608.20114*, 2026. URL https://arxiv.org/abs/2608.20114.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A Large-Scale Simulation Framework for Training and Benchmarking Generalist Robots. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tQJYKwc3n4.

NVIDIA. GR00T N1.5: An improved open foundation model for generalist humanoid robots. https://research.nvidia.com/labs/gear/gr00t-n1_5/, 2025a.

NVIDIA. GR00T N1.6: An improved open foundation model for generalist humanoid robots. https://research.nvidia.com/labs/gear/gr00t-n1_6/, 2025b.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://doi.org/10.1109/ICRA57147.2024.10611477.

Ri-Zhao Qiu, Yafei Hu, Yuchen Song, Ge Yang, Yang Fu, Jianglong Ye, Jiteng Mu, Ruihan Yang, Nikolay Atanasov, Sebastian Scherer, and Xiaolong Wang. Learning generalizable feature fields for mobile manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20952–20959. IEEE, 2025. doi: 10.1109/IROS60139.2025.11246780. URL https://doi.org/10.1109/IROS60139.2025.11246780.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72. PMLR, 2023. URL https://proceedings.mlr.press/v229/rana23a.html.

Florian Seligmann, Emiliyan Gospodinov, Enes Ulas Dincer, and Gerhard Neumann. VLA-FAIL: Efficient task failure detection for finetuned vision-language-action models. *arXiv preprint arXiv:2606.21386*, 2026. URL https://arxiv.org/abs/2606.21386.

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*, 2026. URL https://openreview.net/forum?id=54U3XHf7qq.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10161317. URL https://doi.org/10.1109/ICRA48891.2023.10161317.

Wenhao Sun, Sai Hou, Zixuan Wang, Bo Yu, Shaoshan Liu, Xu Yang, Shuai Liang, Yiming Gan, and Yinhe Han. Dadu-e: Rethinking the role of large language model in robotic computing pipelines. *Journal of Field Robotics*, 43(3):2024–2047, 2025. doi: 10.1002/rob.70120. URL https://doi.org/10.1002/rob.70120.

Priya Sundaresan, Rhea Malhotra, Phillip Miao, Jingyun Yang, Jimmy Wu, Hengyuan Hu, Rika Antonova, Francis Engelmann, Dorsa Sadigh, and Jeannette Bohg. Homer: Learning in-the-wild mobile manipulation via hybrid imitation and whole-body control. *arXiv preprint arXiv:2506.01185*, 2025.

Ruisen Tu, Arth Shukla, Sohyun Yoo, Xuanlin Li, Junxi Li, Jianwen Xie, Hao Su, and Zhuowen Tu. SG-VLA: Learning spatially-grounded vision-language-action models for mobile manipulation. *arXiv preprint arXiv:2603.22760*, 2026. URL https://arxiv.org/abs/2603.22760.

Shagun Uppal, Ananye Agarwal, Haoyu Xiong, Kenneth Shaw, and Deepak Pathak. SPIN: Simultaneous Perception Interaction and Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18133–18142, June 2024. doi: 10.1109/CVPR52733.2024.01717. URL https://openaccess.thecvf.com/content/CVPR2024/html/Uppal_SPIN_Simultaneous_Perception_Interaction_and_Navigation_CVPR_2024_paper.html.

Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. GPT-4V(ision) for Robotics: Multimodal Task Planning From Human Demonstration. *IEEE Robotics and Automation Letters*, 9(11):10567–10574, 2024. doi: 10.1109/LRA.2024.3477090. URL https://doi.org/10.1109/LRA.2024.3477090.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. MimicPlay: Long-Horizon Imitation Learning by Watching Human Play. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 201–221. PMLR, 2023. URL https://proceedings.mlr.press/v229/wang23a.html.

Zixuan Wang, Bo Yu, Junzhe Zhao, Wenhao Sun, Sai Hou, Shuai Liang, Xing Hu, Yinhe Han, and Yiming Gan. KARMA: Augmenting embodied ai agents with long-and-short term memory systems. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8.

IEEE, 2025. doi: 10.1109/ICRA55743.2025.11128047. URL https://doi.org/10.1109/ICRA55743.2025.11128047.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3094–3114. PMLR, 2025. URL https://proceedings.mlr.press/v305/wen25b.html.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026. URL https://arxiv.org/abs/2601.18692.

Zhenyu Wu, Yuheng Zhou, Xiuwei Xu, Ziwei Wang, and Haibin Yan. MoManipVLA: Transferring Vision-Language-Action Models for General Mobile Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1714–1723, June 2025. doi: 10.1109/CVPR52734.2025.00167. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_MoManipVLA_Transferring_Vision-language-action_Models_for_General_Mobile_Manipulation_CVPR_2025_paper.html.

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. URL https://arxiv.org/abs/2606.19980.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, Nan Sun, Qiao Sun, Runze Suo, Heyun Wang, Yunhong Wang, Rujie Wu, Caoyu Xia, Lina Zhang, Jack Zhao, et al. Xiaomi-robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026. URL https://arxiv.org/abs/2607.15330.

Sixu Yan, Zeyu Zhang, Muzhi Han, Zaijin Wang, Qi Xie, Zhitian Li, Zhehan Li, Hangxin Liu, Xinggang Wang, and Song-Chun Zhu. M2Diffuser: Diffusion-based Trajectory Optimization for Mobile Manipulation in 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. doi: 10.1109/TPAMI.2025.3553454. URL https://doi.org/10.1109/TPAMI.2025.3553454.

Donglin Yang, Haoran Chen, Xingyu Chen, Lixing Liu, Manyi Li, Changhe Tu, Ke Xu, Xiaojian Ma, and Si Liu. Learning panorama-aware vla for mobile manipulation with whole-body teleoperation. *arXiv preprint arXiv:2608.02257*, 2026a. URL https://arxiv.org/abs/2608.02257.

Yifan Yang, Zhixiang Duan, Tianshi Xie, Fuyu Cao, Pinxi Shen, Peili Song, Chenyang Zhao, Piaopiao Jin, Guokang Sun, Shaoqing Xu, Yangwei You, and Jingtai Liu. FPC-VLA: A vision-language-action framework with a supervisor for failure prediction and correction. *Expert Systems with Applications*, 316:131742, 2026b. doi: 10.1016/j.eswa.2026.131742. URL https://doi.org/10.1016/j.eswa.2026.131742.

Yifan Ye, Yankai Fu, Yaoxu Lv, Bohan Hou, Jun Cen, Lingdong Kong, Duo Zheng, Tianxing Chen, Jiaming Liu, Ziang Cao, et al. Data pyramid for embodied manipulation. *arXiv preprint arXiv:2607.24744*, 2026.

Zewei Ye, Weifeng Lu, Minghao Ye, Tao Lin, Shuo Yang, Junchi Yan, and Bo Zhao. Robofac: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*, 2025.

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin S. Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander Clegg, John M. Turner, Zsolt Kira, Manolis Savva, Angel X. Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi,

Yonatan Bisk, and Chris Paxton. HomeRobot: Open-Vocabulary Mobile Manipulation. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1975–2011. PMLR, 2023. URL https://proceedings.mlr.press/v229/yenamandra23a.html.

Takuma Yoneda, Jiading Fang, Peng Li, Huanyu Zhang, Tianchong Jiang, Shengjie Lin, Ben Picker, David Yunis, Hongyuan Mei, and Matthew R. Walter. Statler: State-Maintaining Language Models for Embodied Reasoning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15083–15091. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610634. URL https://doi.org/10.1109/ICRA57147.2024.10610634.

Ryan Yu, Pushi Zhang, Starrick Liu, Brae Liu, Miracle Kang, Shalfun Li, Lights Shi, Ellie Ma, Ping Yang, Chris Pan, Jerry Chen, Dongxiu Liu, Rain Sun, Miles Guo, Byron Zhang, Hugo Zhou, Zach Xu, Vincent Chen, Harrison Huang, James Wang, Dance Kuzi, Andy Zhai, Hang Su, Roy Gan, Lucy Liang, Hao Wang, and Qian Wang. Wall-oss-0.5 technical report. *arXiv preprint arXiv:2605.30877*, 2026. URL https://arxiv.org/abs/2605.30877.

Yi Yu and Xinchuan Qiu. Benchmarking vision-language-action models on SO-101: Failure and recovery analysis. *arXiv preprint arXiv:2606.08881*, 2026. URL https://arxiv.org/abs/2606.08881.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, Jie Zhang, Jingyang Fan, Gengze Zhou, Qihang Peng, Chenxu Lv, Xiaoyue Chen, An Yang, Fei Huang, Junyang Lin, Dayiheng Liu, Jingren Zhou, Chenfei Wu, and Xiong-Hui Chen. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026a. URL https://arxiv.org/abs/2606.17846.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026b. URL https://arxiv.org/abs/2603.16666.

Xianchao Zeng, Xinyu Zhou, Youcheng Li, Jiayou Shi, Tianle Li, Liangming Chen, Lei Ren, and Yong-Lu Li. Diagnose, correct, and learn from manipulation failures via visual symbols. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42386–42395, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zeng_Diagnose_Correct_and_Learn_from_Manipulation_Failures_via_Visual_Symbols_CVPR_2026_paper.html.

Xiaopeng Zhang, Yueyang Weng, Qi Liu, Yongjin Mu, and Yanjie Li. Learning robust execution in robotic manipulation with agentic reinforcement learning. *arXiv preprint arXiv:2607.13818*, 2026a. URL https://arxiv.org/abs/2607.13818.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Zhengru Fang, Yi Nie, 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. URL https://arxiv.org/abs/2607.08448v4.

Ganlong Zhao, Zijia Tang, Xingping Chen, Zhanghui Kuang, Ye Tian, and Guanbin Li. FLARE: A failure-aware framework for autonomous correction and recovery in visual-language robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22391–22401, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhao_FLARE_A_Failure-Aware_Framework_for_Autonomous_Correction_and_Recovery_in_CVPR_2026_paper.html.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Qualitative Execution Examples

We provide four representative rollouts to illustrate how the high-level planner composes the fixed subtask VLA with the system-level interventions during long-horizon execution. The examples cover standard multi-stage execution, coordination between navigation and manipulation, accessibility-aware base adjustment, and local end-effector correction after a failed placement.

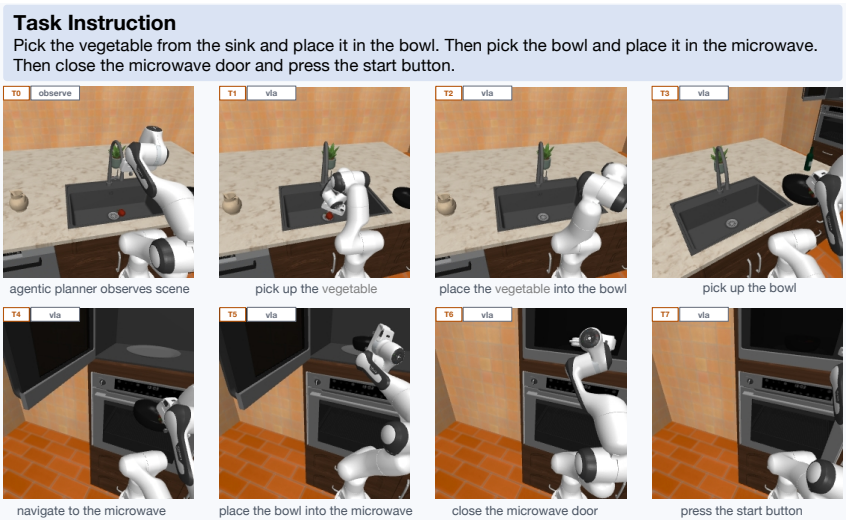

Figure 6: **Multi-stage closed-loop execution.** The high-level planner decomposes the task into a sequence of subtask-VLA calls, including object pickup and placement, navigation to the microwave, interaction with the microwave door, and pressing the start button.

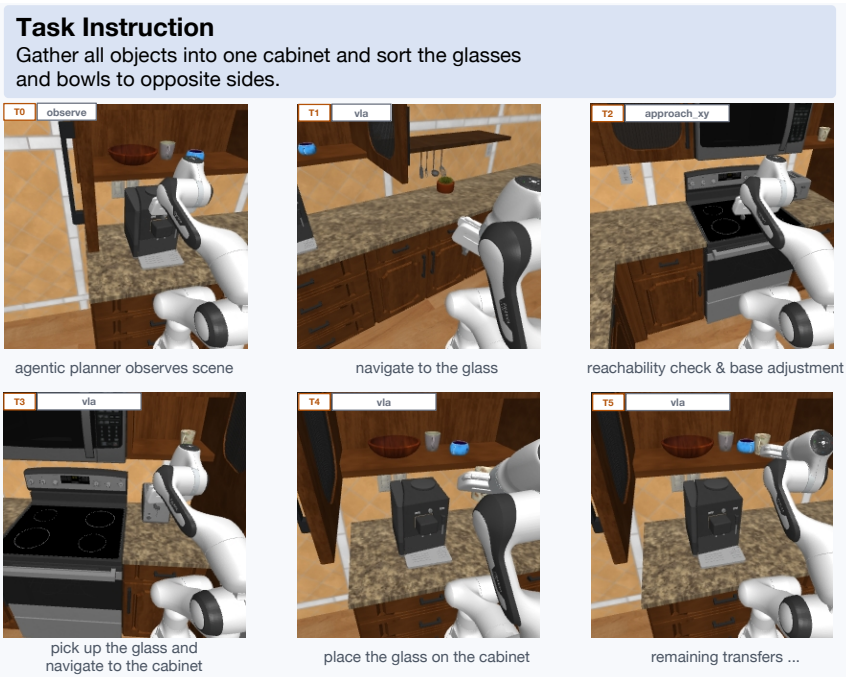

Figure 7: **Accessibility-aware base adjustment before manipulation.** After navigating near the target, the planner checks whether the current base pose is suitable for manipulation and invokes `approach_xy` for a local base adjustment before continuing with the subtask VLA.

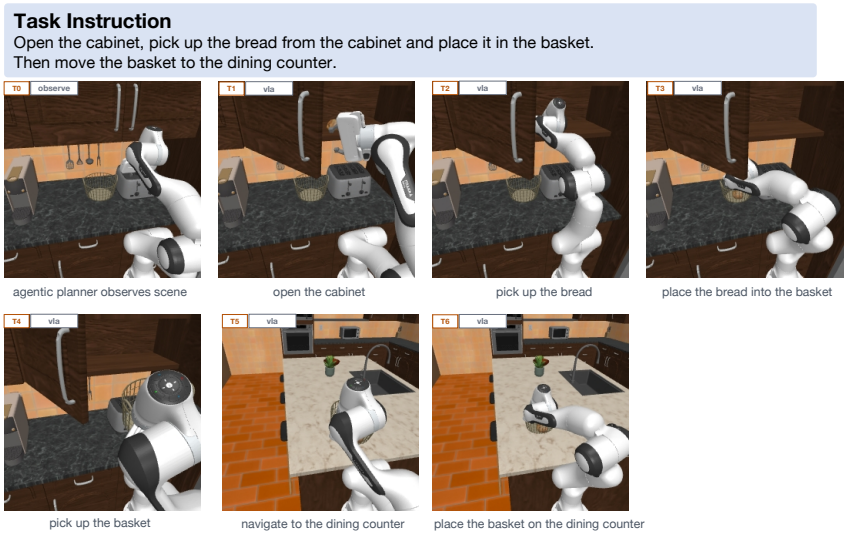

Figure 8: **Coordinated navigation and manipulation in a long-horizon task.** The planner uses the same subtask VLA to open the cabinet, pick and place the bread, transport the basket, navigate to the dining counter, and complete the final placement.

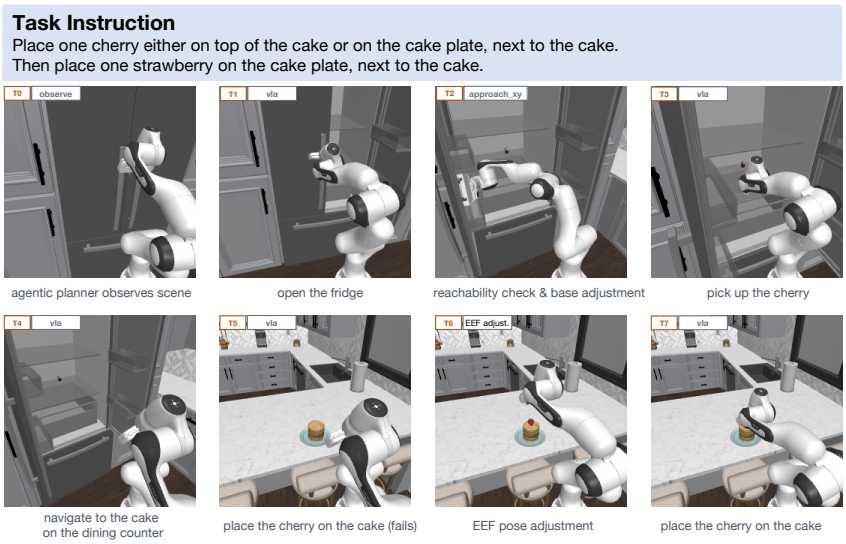

Figure 9: **End-effector adjustment after a failed placement attempt.** After the initial placement fails, the planner invokes a local end-effector adjustment and retries the placement with the same subtask VLA.

B 20-TASK MOBILE MANIPULATION SET

We evaluate all 20 RoboCasa365 benchmark tasks that require mobile manipulation. The tasks cover navigation between workspaces, object pickup and placement, object transport, interaction with articulated objects, and manipulation around appliances and receptacles. Following the official RoboCasa365 decomposition, the tasks contain 2–15 subtasks, with 91 subtasks in total and an average of 4.55 subtasks per task. The task names, subtask counts, and per-task results are reported together in Appendix E to avoid a separate task-list table.

C AGENTIC BASELINE AND SUBTASK VLA DETAILS

C.1 AGENTIC BASELINE

We adapt the tool-based agentic system of Harness VLA (Zhang et al., 2026b). Kimi K3 (AI, 2026) serves as the high-level planner in a closed perception–reasoning–action loop. At each planner step, it either issues a short subtask instruction to the VLA or invokes an available execution tool. Each VLA call executes for 350 control steps before returning control to the planner. The planner then observes the updated scene and decides whether to continue the current subtask, retry it, or proceed to the next one. Navigation and manipulation use the same subtask VLA checkpoint and differ only in the instruction issued by the planner. We retain the planning and closed-loop execution structure of Harness VLA, while removing simulator-specific tools that are not readily transferable to real robots. The same planner interface and execution settings are used across all agentic configurations unless otherwise stated.

C.2 SUBTASK VLA TRAINING

Training data. We train a single subtask-conditioned VLA using only the official RoboCasa365 training split. The training data combine atomic-task demonstrations with subtask level segments extracted from composite task demonstrations. The composite portion contains 16,181 demonstrations and 12.73M frames at 20 Hz. After removing the terminal `done` stage, 12.27M frames remain, yielding 84,401 valid subtask segments. No evaluation trajectory is used for policy training.

Table 3: **Training settings of the subtask VLA.** The same checkpoint is used by all agentic configurations unless otherwise stated.

Setting	Value
VLM backbone	Qwen3-VL-4B-Instruct
Visual encoder	DINOv2 ViT-S/14
Action model	DiT-B
Image resolution	224×224
State dimension	16
Action dimension	12
Action horizon	50
DiT hidden size	1024
DiT layers	16
Inference steps	10
Batch size	256
Training steps	200,000
Warmup steps	1,000
VLM learning rate	2.5×10^{-5}
Action-model learning rate	1.0×10^{-4}
LR scheduler	Cosine
Minimum learning rate	2.5×10^{-6}
Optimizer	AdamW
AdamW β	$(0.9, 0.95)$
Gradient clipping	1.0
Random seed	42

Subtask segmentation. Each frame in a composite demonstration is annotated with a subtask index, a natural-language subtask instruction, an atomic skill label, and an execution stage. We split each trajectory at the annotated subtask boundaries and condition each segment on its corresponding subtask instruction. Action chunks are truncated at segment boundaries and padded with the final action when needed, so a training sample never crosses into the next subtask. The resulting dataset contains 734 distinct subtask instructions.

Model and training. We implement the subtask VLA with StarVLA using the QwenPI architecture. The vision-language backbone is Qwen3-VL-4B-Instruct, together with a DINOv2 ViT-S/14 visual encoder and a DiT-B action head. Each training sample contains three RGB views resized to 224×224, the robot state, a subtask instruction, and a 50-step action chunk. The policy predicts the 12-dimensional RoboCasa action space from a 16-dimensional robot state. The action head uses a hidden size of 1024, 16 DiT layers, and 10 inference steps.

Atomic-task and subtask-level samples are jointly trained in a single checkpoint. We balance the two data sources so that frames in the combined dataset have the same long-run sampling probability, and sample without replacement within each epoch. Navigation and manipulation use the same checkpoint and differ only in the subtask instruction. The trained model achieves 80% success on the RoboCasa365 atomic-task evaluation.

End-to-end baseline checkpoints. For the end-to-end baselines in Table 1, we use the released checkpoints and corresponding implementations linked from the official RoboCasa365 leaderboard. Table 4 summarizes the training settings reported by the corresponding submissions. We evaluate these released checkpoints on the same 20 mobile manipulation tasks and matched episode seeds used in our evaluation.

Table 4: **Training settings of the released end-to-end baselines.** Batch sizes and training steps follow the corresponding RoboCasa365 leaderboard submissions.

Method	Batch Size	Training Steps
π_0	64	75,000
$\pi_{0.5}$	64	75,000
GR00T N1.5	128	120,000
GR00T N1.6	128	120,000
RLDX-1	192	250,000
Xiaomi Robotics 1	512	120,000

D FAILURE ATTRIBUTION DETAILS

D.1 ATTRIBUTION PROCEDURE AND ANNOTATION CONSISTENCY

Offline diagnosis. We attribute failures using information recorded during each rollout. For every failed episode, the diagnostic input includes the rollout video, agent interaction history, robot states, and simulator object states. Simulator states are used only for offline diagnosis and are never exposed to the online agent. We use Codex (GPT-5.6 Sol) to inspect each failed rollout. For each episode, we run the same diagnosis prompt three times and obtain three candidate failure attributions. The resulting diagnoses are then jointly reviewed by three human reviewers to determine the final failure label. The primary failure follows the definition in Section 3.2: we label the first error that affects the final task outcome and is not later recovered.

Diagnosis prompt. The same diagnosis prompt is used for all failed episodes. The prompt asks the model to identify the first unrecovered error, rather than downstream consequences, and to assign it to one of the predefined failure categories. We reproduce the exact prompt used in our experiments below.

Diagnosis prompt. The same diagnosis template is used for all failed episodes. The model is asked to identify the first unrecovered error, rather than downstream consequences, and assign it to the predefined failure taxonomy.

```
You are given a failed long-horizon mobile manipulation
episode.  Diagnose the primary cause of failure using the
task instruction, rollout video, agent interaction history,
robot states, and simulator object states.
Your goal is to identify the first unrecovered error that
affects the final task outcome.
Follow these rules:
1.  Inspect the episode in temporal order.
2.  Find the earliest step at which an error occurs that
contributes to the final task failure and is not later
recovered.  Treat this step as the primary failure point.
3.  Do not label downstream consequences as separate primary
failures.  For example, if an incorrect subtask causes the
robot to reach the wrong workspace and all later actions
fail, attribute the episode to the original planning error.
4.  Assign the primary failure to exactly one of the
following categories:
Task planning:  - Placement target or receptacle assignment
error - Subtask ordering error
Navigation and base positioning:  - Ineffective approach or
poor base positioning - Collision, path blockage, or stalled
navigation - Unsuitable base orientation
Manipulation:  - Incorrect object placement - Grasp
misalignment or empty grasp - Unstable grasp or slippage
- Collision-induced scene disruption - Object dropped during
transport - Articulated-object manipulation failure
5.  Base the diagnosis only on evidence available in the
provided episode.  Use simulator object states only to
verify what happened during offline diagnosis; do not assume
that this information was available to the online agent.
Return:
Primary failure point:  Briefly identify when the first
unrecovered error occurs.
Primary failure category:  Choose one of Task planning,
Navigation and base positioning, or Manipulation.
Fine-grained failure category:  Choose one category from the
list above.
Evidence:  Briefly describe the observation or state change
that supports the attribution, and explain why any later
errors are consequences rather than the primary failure.
```

Primary failure attribution consistency. To examine the consistency of the primary failure attribution, we randomly sample 30 failed episodes from the agentic baseline. Three reviewers independently assign each episode to one of the three primary failure categories used in Section 3.2: task planning, navigation and base positioning, or manipulation.

The reviewers agree on 26 of the 30 episodes (86.7%), with a Fleiss' κ of 0.86. The remaining four cases are resolved by joint review before obtaining the final labels. Table 5 summarizes the annotation results.

Fine-grained attribution consistency. We further evaluate the consistency of the fine-grained failure labels used in Figure 2. We randomly sample 30 failed episodes across the evaluated agentic configurations. For this evaluation, the system configuration and enabled skills are hidden from the reviewers.

Three reviewers independently assign each episode to one of the fine-grained failure categories based on the rollout video, agent history, robot states, and simulator object states. The reviewers unanimously agree on 24 of the 30 episodes (80.0%), with a Fleiss' κ of 0.79. The blinded majority label also agrees with the original fine-grained attribution on 27 of the 30 episodes (90.0%).

Table 5: **Agreement of independent failure attribution on a random subset of baseline failures.**

Failure Category	Episodes	Unanimous
Task planning	7	6
Navigation and base positioning	8	7
Manipulation	15	13
Total	30	26

The remaining disagreements mainly occur between closely related execution failures, such as incorrect placement and grasp or transport failures. Overall, the fine-grained labels remain consistent when the system configuration is hidden from the reviewers.

D.2 Paired Failure Transitions

To complement the aggregate failure counts in Figure 5, we track the outcome of each matched episode from the baseline to the full system. Table 7 shows transitions among success, task-planning failure, navigation/base-positioning failure, and manipulation failure.

The transition table makes this survivor effect explicit. Six of the 17 baseline planning failures and six of the 21 navigation/base-positioning failures become manipulation failures in the full system. These transitions show that part of the increase in manipulation failures comes from episodes reaching later stages after earlier failures are addressed. They do not determine whether the underlying manipulation failure rate also changed. The transition counts also separate gains from regressions. The full system recovers 30 of the 81 baseline failures (37.0%) and retains 17 of the 19 baseline successes (89.5%). Two previously successful episodes become failures (10.5% of baseline successes). Among the baseline failure categories, recovery is 9/17 for planning (52.9%), 12/21 for navigation/base positioning (57.1%), and 9/43 for manipulation (20.9%). These are conditional on the baseline episode label, not on entering a particular execution stage. They therefore do not estimate grasp or placement success, nor do they assign recovery to a single tool.

D.3 Fine-Grained Failure Attribution of the Full System

Among the 53 failed full-system episodes, 47 are attributed to manipulation, five to navigation and base positioning, and one to task planning. The remaining manipulation failures are mainly incorrect object placement, object drops during transport, and collision-induced scene disruption. This distribution is consistent with the bottleneck shift discussed in the main paper.

E Per-Task and Horizon Results

Table 8 combines the task list, official subtask counts, and full-system results on the development episodes. Each task is evaluated with the same five episode seeds used in the main evaluation.

We also group tasks by the number of annotated subtasks to provide a descriptive view of performance across task horizons. The full system improves over the baseline in both groups. Observed success is lower for tasks with 5–15 subtasks than for tasks with 2–4 subtasks in both systems.

Reference from error compounding. The subtask VLA achieves 80% success on the atomic-task evaluation. As a simple reference for the effect of task length, we treat each annotated subtask as an independent event with success probability $p = 0.8$. For a task with H_i subtasks, the reference success rate is 0.8^{H_i}. Averaging over the 20 tasks gives

$$\frac{1}{20}\sum_{i=1}^{20} 0.8^{H_i} = 41.0\%.$$

The corresponding averages are 49.6% for the 13 tasks with 2–4 subtasks and 25.0% for the 7 tasks with 5–15 subtasks. The observed agentic baseline achieves 19.0%.

Table 6: **Agreement of fine-grained failure attribution.** Three reviewers independently annotate 30 sampled failed episodes without access to the system configuration or enabled skills.

Fine-grained failure category	Episodes	Unanimous
Placement target or receptacle assignment	4	3
Subtask ordering	3	2
Ineffective approach or poor base positioning	4	4
Collision, path blockage, or stalled navigation	3	2
Unsuitable base orientation	2	2
Incorrect object placement	5	4
Grasp misalignment or empty grasp	3	3
Unstable grasp or slippage	2	1
Collision-induced scene disruption	2	1
Object dropped during transport	1	1
Articulated-object manipulation failure	1	1
Total	30	24

Table 7: Paired episode-level outcome transitions from the agentic baseline to the full system.

Baseline → Full	Success	Planning	Nav./Base	Manip.	Total
Success	17	0	1	1	19
Planning	9	1	1	6	17
Nav./Base	12	0	3	6	21
Manipulation	9	0	0	34	43
Total	47	1	5	47	100

F Leave-One-Out Ablation of Intervention Groups

The progressive integration in Figure 4 follows the order motivated by the baseline failure analysis, so the incremental gain of a component can depend on which components are already present. We therefore perform a leave-one-out ablation from the full system. Each configuration removes one intervention group while keeping all other components unchanged and uses the same 100 matched development episodes.

Removing base adjustment produces the largest observed drop, from 47.0% to 35.0%, consistent with its large gain in the progressive experiment. Removing the other intervention groups also reduces the success rate. Paired confidence intervals and McNemar tests for these comparisons, together with the held-out leave-one-out results, are reported in Appendix I.

G Cross-benchmark evaluation on five BEHAVIOR-1K tasks

To evaluate the system design beyond RoboCasa365, we test the framework on five BEHAVIOR-1K 2026 Challenge tasks: Pick Up Trash, Make Rose Centerpieces, Composting Waste, Installing Smoke Detectors, and Make Microwave Popcorn.

For this evaluation, we train the low-level policies using only the training data provided by the BEHAVIOR-1K challenge. Following the RoboCasa365 setup, our subtask VLA is trained from trajectories with the provided subtask annotations: each trajectory is segmented according to its subtask boundaries and conditioned on the corresponding subtask instruction. In contrast, the $\pi_{0.5}$ baseline is trained on the corresponding complete trajectories without subtask-level supervision. For the agentic methods, Kimi K3 is used as the high-level planner. Each method is evaluated for 20 rollouts per task, resulting in 100 rollouts in total. Corresponding rollouts across methods use the same scene seed and identical initial simulator and object states.

As shown in Table 11, the full system consistently improves over Harness VLA across all five tasks, increasing the overall success rate from 14.0% to 37.0%, while the end-to-end model $\pi_{0.5}$ achieves 7.0%. This experiment evaluates transfer of the agentic system design, with the low-level policies retrained on BEHAVIOR-1K training data.

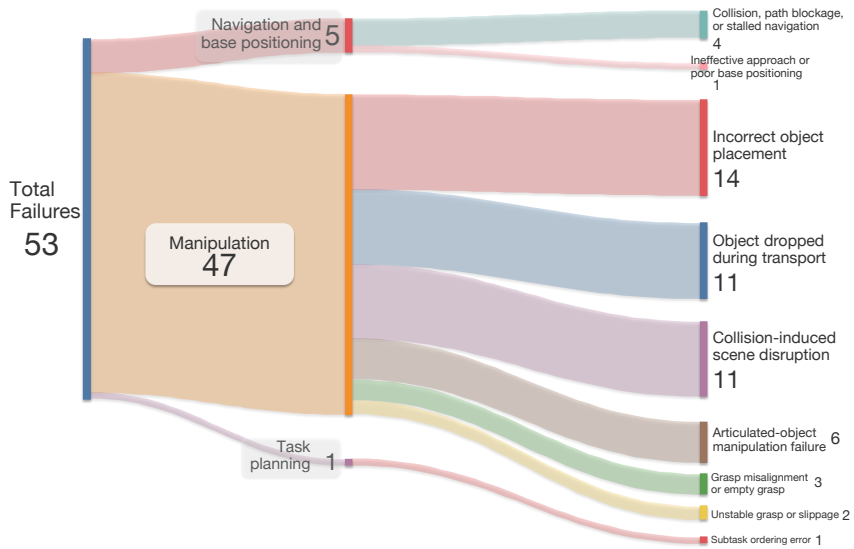

Figure 10: **Fine-grained failure attribution for the full system.** Counts are reported over the same 100 matched development episodes.

Table 8: **Per-task results on the 20 mobile manipulation tasks of RoboCasa365.** The number of subtasks follows the official RoboCasa365 decomposition. Each task is evaluated with five matched development episode seeds.

Task	# Subtasks	Full	Task	# Subtasks	Full
DeliverStraw	4	1/5	ArrangeBreadBasket	5	2/5
GetToastedBread	4	2/5	BreadSelection	2	3/5
PackIdenticalLunches	15	1/5	GarnishPancake	4	4/5
ScrubCuttingBoard	2	4/5	GatherTableware	4	2/5
SearingMeat	3	2/5	HeatKebabSandwich	6	0/5
SetUpCuttingStation	2	2/5	MakeIceLemonade	5	3/5
StackBowlsCabinet	2	5/5	PortionHotDogs	4	4/5
SteamInMicrowave	6	3/5	RecycleBottlesByType	3	4/5
StirVegetables	4	1/5	SeparateFreezerRack	7	2/5
StoreLeftoversInBowl	5	1/5	WaffleReheat	4	1/5
Overall: 47/100 (47.0%)					

H Additional Skill Implementation Details

This appendix provides additional implementation details for the skill library, adaptive visual memory, and geometric skills used in the main experiments.

Table 9: **Performance across task horizons.** Tasks are grouped by the number of annotated subtasks.

Task horizon	# Tasks	Baseline	Full system	Δ (pp)
2–4 subtasks	13	15/65 (23.1%)	35/65 (53.8%)	+30.7
5–15 subtasks	7	4/35 (11.4%)	12/35 (34.3%)	+22.9
Overall	20	19/100 (19.0%)	47/100 (47.0%)	+28.0

Table 10: **Leave-one-out ablation of intervention groups on the development episodes.**

Configuration	Task Success Rate(%)	Δ from Full (pp)
Full system	**47.0**	–
− Skill library	39.0	-8.0
− Adaptive visual memory	41.0	-6.0
− Base adjustment	35.0	**-12.0**
− End-effector adjustment	40.0	-7.0

Table 11: Cross-benchmark evaluation on five BEHAVIOR-1K 2026 Challenge tasks. Each method is evaluated on 20 matched episodes per task, using the same scene seeds and initial states across methods. Overall results are aggregated over 100 episodes per method.

Method	Pick Up Trash	Make Rose Centerpieces	Composting Waste	Installing Smoke Detectors	Make Microwave Popcorn	Overall
$\pi_{0.5}$	1 (5%)	2 (10%)	1 (5%)	1 (5%)	2 (10%)	7 / 100 (7.0%)
Harness VLA	2 (10%)	3 (15%)	3 (15%)	2 (10%)	4 (20%)	14 / 100 (14.0%)
Ours	**7 (35%)**	**9 (45%)**	**7 (35%)**	**6 (30%)**	**8 (40%)**	**37 / 100 (37.0%)**

H.1 Demonstration Retrieval and Skill Cards

For each evaluation task, we retrieve a demonstration from the RoboCasa365 benchmark training split. The retrieval query uses the task instruction, and the candidate demonstrations are compared using their task annotations. After retrieval, Gemini 3.5 Flash summarizes the demonstration into a skill card containing the task goal, ordered subtask plan, representative keyframes, and completion criteria. Skill cards are generated once before evaluation and remain fixed during rollout. The planner receives the skill card together with the current observation and execution history when planning the next step. No evaluation trajectory, evaluation outcome, or simulator state is used for demonstration retrieval or skill-card generation.

H.2 Adaptive Visual Memory

When the planner calls `recall_memory(q)`, the recall model receives the natural-language query together with the stored candidate frames. It ranks the frames according to their relevance to the query and returns the top-k frames together with a short summary of the corresponding trajectory history. We use $k = 16$ in the full system unless otherwise stated in the ablation study.

The recall model uses Qwen3.8-27B, and candidate frames are resized to 448×448 before being passed to the model. Recall only operates on observations collected before the current agent step. We keep at most 256 candidate frames for each episode. When more than 256 observations are available, we uniformly subsample the past observations to form the candidate set. The recall model ranks these candidates conditioned on the agent query, and the top-k frames are returned to the planner. Adaptive recall means that the planner decides when to call the recall tool and provides a query based on its current information need; fixed recall instead provides uniformly sampled history frames without an agent-generated query.

H.3 Geometric Skill Parameters

The geometric skills use the RGB-D observations available to the online agent. Grounding DINO first localizes the object or interaction region specified by the current subtask, and SAM refines the corresponding mask. Valid depth pixels are then back-projected into the robot coordinate frame.

For the accessibility check, a target within 0.75 m of the robot is classified as reachable. Targets between 0.75 m and 1.00 m are classified as marginal, while targets beyond 1.00 m are treated as out of reach. For a marginal stance, `approach_xy` applies a local base correction of at most 0.25 m while keeping the base heading fixed. The 0.75 m and 1.00 m accessibility thresholds are estimated from successful trajectories in the training split and are fixed before evaluation.

Table 12: Implementation settings for adaptive visual memory and geometric skills.

Parameter	Value
Adaptive visual memory	
Maximum candidate frames	256
Retrieved frames (k)	16
Recall model	Qwen3.8-27B
Input image resolution	448×448
Geometric skills	
Reachable distance: $d \leq 0.75\,\mathrm{m}$	
Marginal distance: $0.75 < d \leq 1.00\,\mathrm{m}$	
Maximum local XY shift	0.25 m
Obstacle inflation	0.10 m
Maximum EEF correction	0.30 m
Occupancy-grid resolution	0.05 m

For an out-of-reach stance, `goto_stance` constructs a 2D occupancy grid from the current depth observation and searches for a new base pose within the manipulation range. The occupancy grid uses a resolution of 0.05 m. Obstacles are inflated by 0.10 m before A* planning.

For end-effector adjustment, the median 3D position of the grounded region is used as the target position. The local surface normal is estimated from the same point set and used to define the approach direction. Each adjustment is bounded to a maximum translation of 0.30 m. Table 12 summarizes the main settings used by the adaptive visual memory and geometric skills.

H.4 Planner prompt excerpts

The following summary covers the planner instructions for task execution, demonstration use, adaptive visual memory, and local correction.

> **Task execution.** Read the task instruction and current observations. Issue one short subtask instruction per VLA call. After execution, inspect the returned observations before choosing the next subtask. A completed tool call does not by itself mean that the subtask is complete.
>
> **Demonstration guidance.** When a skill card is provided, use its plan and keyframes as a reference. Compare the current observations with the expected outcome before continuing. The demonstration scene may differ from the current scene; use the demonstrated behavior as guidance rather than copying object coordinates.
>
> **Adaptive visual memory.** Query memory when past observations are needed to understand task progress or a failed attempt. Retrieved frames describe earlier states; use the latest observations to determine the current scene state.
>
> **Failure recovery.** After an empty grasp or an off-target placement, inspect what happened before retrying. A closed gripper alone does not confirm a successful grasp. Localize the target from the current RGB-D observations rather than using fixed coordinates. Use local end-effector correction to address misalignment, then retry with the subtask VLA.
>
> **Tool calls.** Briefly state the relevant observation and the chosen action before a tool call. Do not request another observation when the preceding tool has already returned an updated one.

I Paired Statistical Analysis

All progressive integration, leave-one-out, and fine-grained ablation experiments use matched episode seeds and simulator initial states across configurations. We therefore use episode-level paired analyses rather than treating success rates from two configurations as independent binomial samples.

For configurations A and B, let n_{01} be the number of matched episodes that fail under A and succeed under B, and let n_{10} be the number that succeed under A and fail under B. The paired change in task

success rate is

$$\Delta_{\text{succ}} = \frac{n_{01} - n_{10}}{N},$$

where $N = 100$ for the RoboCasa365 evaluations. We use a two-sided exact McNemar test for paired binary outcomes. Confidence intervals for the paired success-rate difference are estimated with paired bootstrap resampling stratified by task, preserving the 20 task, five seed structure.

I.1 Progressive Integration

Table 13: **Paired analysis of progressive integration.** The first four rows compare consecutive configurations; the final row compares the baseline with the full system on the same development episodes. Δ is reported in percentage points.

Comparison	Success (%)	Δ (pp)	n_{01}	n_{10}	95% CI (pp)	McNemar p
Baseline $\rightarrow$ +Demo	19 $\rightarrow$ 25	+6	8	2	[0, 12]	0.109
+Demo $\rightarrow$ +Memory	25 $\rightarrow$ 29	+4	7	3	[-2, 10]	0.344
+Memory $\rightarrow$ +Base	29 $\rightarrow$ 40	+11	15	4	[3, 19]	0.019
+Base $\rightarrow$ +EEF	40 $\rightarrow$ 47	+7	10	3	[0, 14]	0.092
Baseline $\rightarrow$ Full	19 $\rightarrow$ 47	+28	30	2	[18, 38]	< 0.001

The baseline-to-full comparison shows a large paired improvement of 28 percentage points. Among the individual integration steps, base adjustment has the clearest paired effect. The smaller changes from skill library, adaptive visual memory, and end-effector adjustment have wider confidence intervals, so we treat them as supporting component evidence rather than separate strong statistical claims.

I.2 Leave-One-Out and Held-Out Evaluation

Table 14: **Paired leave-one-out analysis on development and held-out episodes.** "Memory" denotes adaptive visual memory.

Split	Comparison	Success (%)	Δ (pp)	n_{01}	n_{10}	95% CI (pp)	McNemar p
Development	$-$Demo $\rightarrow$ Full	39 $\rightarrow$ 47	+8	11	3	[1, 15]	0.057
	$-$Memory $\rightarrow$ Full	41 $\rightarrow$ 47	+6	9	3	[-1, 13]	0.146
	$-$Base $\rightarrow$ Full	35 $\rightarrow$ 47	+12	14	2	[5, 20]	0.004
	$-$EEF $\rightarrow$ Full	40 $\rightarrow$ 47	+7	10	3	[0, 14]	0.092
Held-out	Baseline $\rightarrow$ Full	18 $\rightarrow$ 49	+31	33	2	[21, 41]	< 0.001
	$-$Demo $\rightarrow$ Full	40 $\rightarrow$ 49	+9	12	3	[2, 16]	0.035
	$-$Memory $\rightarrow$ Full	44 $\rightarrow$ 49	+5	9	4	[-2, 12]	0.267
	$-$Base $\rightarrow$ Full	38 $\rightarrow$ 49	+11	14	3	[3, 19]	0.013
	$-$EEF $\rightarrow$ Full	42 $\rightarrow$ 49	+7	11	4	[0, 14]	0.118

The leave-one-out results show the same overall pattern on development and held-out episodes. Base adjustment has the largest leave-one-out effect in both splits. Most importantly, after the system design, prompts, retrieval settings, and tool parameters are fixed, the full system improves from 18.0% to 49.0% over the agentic baseline on held-out episodes, with a paired difference of 31 percentage points and $p < 0.001$.

I.3 Failure Category of Recovered Episodes

To check whether each intervention mainly addresses the failure category that motivated it, we inspect matched episodes that change from failure to success when the intervention is added. Table 15 reports the primary failure category before the intervention.

The recovered episodes largely match the intended failure-to-intervention mapping: skill library and adaptive visual memory mainly recover task-planning failures, base adjustment mainly recovers navigation and base-positioning failures, and end-effector adjustment mainly recovers manipulation

Table 15: **Primary failure types of episodes recovered after each intervention.**

Added intervention	Recovered	Planning	Nav./Base	Manip.
Skill library	8	5	1	2
Adaptive visual memory	7	4	1	2
Base adjustment	15	0	11	4
End-effector adjustment	10	0	1	9

failures. An intervention can still affect later stages of a long-horizon episode, so this analysis should be interpreted at the episode level rather than as a one-to-one causal decomposition.

I.4 Fine-Grained Ablations

The fine-grained ablations in Table 2 also use matched episode seeds and simulator initial states. Table 16 reports the paired comparisons.

Table 16: **Paired analysis of the fine-grained ablations in Table 2.** All comparisons use the same 100 matched development episodes.

Comparison	Success (%)	Δ (pp)	n_{01}	n_{10}	95% CI (pp)	McNemar p
Skill library						
HIW-500 $\rightarrow$ in-domain video	40 $\rightarrow$ 47	+7	12	5	[-1, 15]	0.143
Text-only $\rightarrow$ full skill card	43 $\rightarrow$ 47	+4	7	3	[-2, 10]	0.344
Adaptive visual memory						
Uniform-8 $\rightarrow$ Adaptive-8	37 $\rightarrow$ 42	+5	8	3	[-1, 11]	0.227
Uniform-16 $\rightarrow$ Adaptive-16	40 $\rightarrow$ 47	+7	10	3	[0, 14]	0.092
Uniform-32 $\rightarrow$ Adaptive-32	35 $\rightarrow$ 45	+10	13	3	[2, 18]	0.021
Adaptive-8 $\rightarrow$ Adaptive-16	42 $\rightarrow$ 47	+5	9	4	[-2, 12]	0.267
Adaptive-32 $\rightarrow$ Adaptive-16	45 $\rightarrow$ 47	+2	6	4	[-4, 8]	0.754
Base adjustment						
w/o `check_stance` $\rightarrow$ Full	36 $\rightarrow$ 47	+11	14	3	[3, 19]	0.013
w/o `approach_xy` $\rightarrow$ Full	42 $\rightarrow$ 47	+5	10	5	[-2, 12]	0.302
w/o `goto_stance` $\rightarrow$ Full	39 $\rightarrow$ 47	+8	12	4	[0, 16]	0.077
End-effector adjustment						
Planner target $\rightarrow$ geometric target	43 $\rightarrow$ 47	+4	9	5	[-3, 11]	0.424

The paired results are consistent with the descriptive ablations in the main paper. Adaptive memory has higher observed success than the corresponding uniform-history setting at all three frame budgets, although the uncertainty remains wide for several comparisons. Within base adjustment, removing the accessibility check produces the clearest drop. The difference between planner-specified and geometrically estimated end-effector targets is smaller and its confidence interval includes zero.

I.5 Separating Memory Access and Frame Selection

The visual-memory ablation in Table 2 changes two factors at once: when memory is accessed and which history frames are returned. We separate these factors using the same 16-frame budget. Memory is either provided at every planner turn or accessed only when requested by the planner. Within each setting, frames are either uniformly sampled from the episode history or ranked by their relevance to the retrieval query. All four configurations use the same 100 matched development episodes.

For planner-triggered uniform memory, the planner decides when memory is needed, but the returned frames are uniformly sampled from the available episode history. For every-turn relevance-based memory, retrieval is performed at every planner turn, using a query generated from the current task state and execution history to rank past observations. All other memory settings, including the candidate history, frame budget, and returned trajectory summary, are kept unchanged.

Table 17: **Separating memory access and frame selection.** All configurations use a 16-frame memory budget and the same 100 matched development episodes.

Memory Access	Uniform	Relevance-based
Every turn	40.0	45.0
Planner-triggered	44.0	**47.0**

Relevance-based frame selection improves the observed success rate under both access settings, while planner-triggered access also improves over providing memory at every turn. Combining planner-triggered access with relevance-based retrieval gives the highest observed success rate. These results suggest that both when memory is accessed and which observations are retrieved contribute to the adaptive-memory design.

J REAL-ROBOT EVALUATION

We further evaluate the framework on Astribot-S1, a wheeled mobile dual-arm robot equipped with parallel grippers. We consider two real-robot tasks, *Storage Box Transport* and *Table Service*.

Tasks. For *Storage Box Transport*, the robot starts away from a cabinet and must move a storage box from the cabinet to a table. The high-level task instruction is:

> *Move the storage box from the cabinet to the table.*

The task contains three main subtasks:

1. Go to the cabinet and pick up the storage box.
2. Move to the table with the storage box.
3. Place the storage box on the table.

For *Table Service*, the robot serves a prepared meal from a serving cart to a dining table. The high-level task instruction is:

> *Serve the meal at the dining table by setting out the prepared dish, adding the topping, and placing the wine glass beside it.*

The task contains four main subtasks:

1. Move the prepared dish from the serving cart to the dining table.
2. Add the topping from the bowl to the prepared dish.
3. Place the bowl on the serving cart.
4. Pick up the the wine glass and place it beside the dish.

Storage Box Transport requires navigation before pickup, object transport during navigation, and placement at a second workspace. Table Service contains multiple manipulation stages.

VLA training and agent setup. For both tasks, we initialize the VLA from the released $\pi_{0.5}$ checkpoint and fine-tun it on real-robot demonstrations. We collect 500 demonstrations for Storage Box Transport and 300 demonstrations for Table Service.

The end-to-end VLA and the agentic methods use different forms of supervision from the same real-robot demonstrations. For the end-to-end VLA baseline, $\pi_{0.5}$ is fine-tuned directly on complete trajectories conditioned on the high-level task instruction. For the agentic methods, we first use a VLM (Gemini 3.5 flash) to convert the demonstrations into subtask-level training data. Each segment is paired with a subtask instruction, and $\pi_{0.5}$ is then fine-tuned on these subtask-level examples. The resulting policy is used as the subtask VLA for both the agentic baseline and the full system. Kimi K3 serves as the high-level planner for both agentic methods, following the simulation experiments.

The planner selects a subtask instruction for the subtask VLA based on the current observation and execution history. The agentic baseline uses the planner together with the subtask VLA. The full system further uses demonstration-based planning guidance, adaptive visual memory, base adjustment, and end-effector adjustment.

Evaluation. We compare three settings: the end-to-end $\pi_{0.5}$ VLA, the agentic baseline, and the full system. Each method is evaluated over 20 real-robot trials.

Table 18: **Real-robot evaluation.** Each method is evaluated for 10 trials on each task.

Method	Storage Box Transport	Table Service	Overall
$\pi_{0.5}$ (end-to-end VLA)	2/10	2/10	4/20
Agentic baseline	2/10	3/10	5/20
Full system	**6/10**	**7/10**	**13/20**

Across the two tasks, the full system succeeds in 13/20 trials, compared with 5/20 for the agentic baseline. The improvement is observed on both tasks rather than being concentrated in a single task. These results follow the same overall trend as the simulation evaluation, where the added interventions improve long-horizon task completion around the same agentic structure.

Qualitative example. Figure 11 shows a representative successful rollout of the Table Service task. The robot first transfers the prepared dish from the serving cart to the dining table, adds the topping, and then places the wine glass beside the dish.

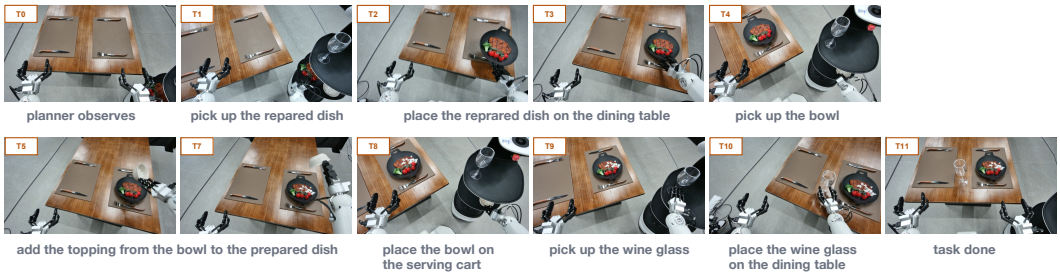

Figure 11: **Real-robot execution of the Table Service task.** A representative successful rollout of the full system on Astribot-S1. The robot moves the prepared dish from the serving cart to the dining table, adds the topping, and places the wine glass beside the dish.