Agentic Decomposition for Reasoning-Oriented Real-World Robot Manipulation

Yutong Lin[1,2] Zhenxuan Fan[1] Yuqian Yuan[1] Wentong Li[3]
Wenqiao Zhang[1,†] Changxu Cheng[2] Tao Wang[2] Juncheng Li[1]
Jun Xiao[1] Siliang Tang[1] Yueting Zhuang[1]
[1]Zhejiang University [2]Uni-Ubi
[3]Nanjing University of Aeronautics and Astronautics
[†]Corresponding author. 2982686152@qq.com

Abstract

Generalist vision-language-action (VLA) models provide strong visuomotor priors, but reasoning-oriented robot manipulation also requires task interpretation, scene grounding, atomic skill decomposition, progress monitoring, and recovery. We introduce AgentVLA, an agentic task interface that treats a VLA model as a reusable low-level executor inside a tool-augmented robot system. AgentVLA converts multi-view observations and task instructions into scene-grounded atomic-skill context, invokes tools for grounding and diagnosis, and conditions the executor on explicit task-stage information while keeping the robot-control protocol unchanged. We validate this design on the ManipArena Challenge at the CVPR 2026 Embodied AI Workshop, where our system achieved the highest success rate in the preliminary evaluation across five bimanual manipulation tasks. Our analysis further shows that noisy demonstrations can teach premature task completion, while spatial-layout shifts can induce brittle geometric priors. These failures motivate separating task reasoning from low-level action generation by atomizing long-horizon tasks into grounded, reusable, and verifiable skill primitives around a shared VLA executor.

1. Introduction

Vision-language-action (VLA) models provide a compelling abstraction for robot manipulation: a policy consumes visual observations, proprioception, and language, and produces continuous actions [1–3, 6–8]. World-action models offer a complementary paradigm by predicting future visual states and actions [11]. Both lines provide strong visuomotor priors, but reasoning-oriented manipulation is not only a control problem. The robot must preserve task state: which object remains unhandled, which relation defines success, when a subgoal is complete, and whether recovery is needed. If these decisions are folded into a single imitation policy, noisy demonstrations can teach premature termination and narrow spatial-layout distributions can become brittle geometric priors. This motivates an agentic decomposition that atomizes long-horizon tasks into grounded, reusable, and verifiable skill primitives before delegation to a VLA executor.

Recent agentic VLA systems make a related observation: a VLA need not be the entire robot agent. Agentic Robot [10] augments a VLA executor with planning, verification, and recovery, while VLA^2 [12] uses retrieval, grounding, masking, and language replacement to convert unseen concepts into executor-friendly representations. We study a complementary real-world setting where objects can remain in distribution, but failures still arise because progress, completion conditions, spatial relations, and skill choices are weakly represented. We frame this as a *task-interface* problem: exposing scene-grounded task structure to a pretrained VLA executor while keeping the low-level action protocol unchanged.

2. Method

AgentVLA treats the VLA as a general-purpose visuomotor executor inside a tool-augmented robot policy (Fig. 1). The task interface converts raw instructions into compact scene-grounded plans, makes target relations and atomic skill order explicit, conditions the executor on the current skill stage, and records full-chain diagnostics so closed-loop failures can be traced to perception, language, or control.

In our implementation, an embodied foundation model planner [5] produces compact plans from multi-view observations and task instructions. The plan is not a symbolic controller and does not replace visuomotor control; it is a structured conditioning signal. The executor is trained and deployed with the same interface: three RGB views, robot state, and the generated plan. We use a $\pi_{0.5}$-style executor [8]; the same interface can be used with other open VLA executors such as OpenVLA [6]. For desktop manipulation, we use the official 14D end-effector representation: left-arm 7D and right-arm 7D end pose. End-effector pose dimensions are trained as deltas while grip-

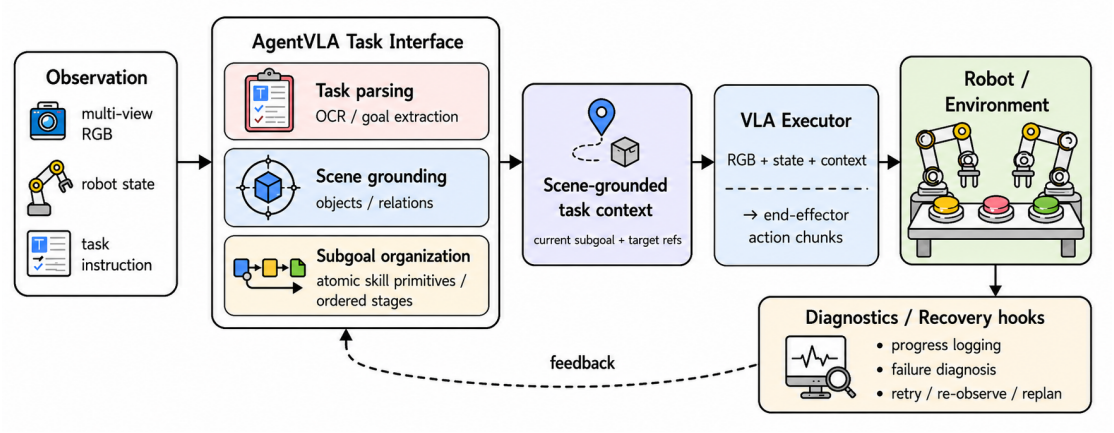

Figure 1. AgentVLA architecture. The agent parses instructions, grounds scene entities, organizes skill stages, and delegates end-effector action chunks to a reusable VLA executor.

per values remain absolute; deployment converts predicted chunks back to absolute end-pose trajectories. This chunked continuous-control formulation is related to recent visuomotor action-generation policies [4]; our focus is the task interface around the executor.

3. Experiments and Discussion

We evaluate on ManipArena [9], a real-robot challenge for reasoning-oriented generalist manipulation at the CVPR 2026 Embodied AI Workshop. In its preliminary evaluation, our system ranked first in success rate across five real desktop tasks. Table 1 summarizes the official results.

Task	$\pi_{0.5}$ Single	$\pi_{0.5}$ OneModel	DreamZero	AgentVLA (Ours)
put_ring	41 / 20%	60 / 20%	27 / 0%	**93 / 90%**
put_spoon	52 / 20%	43 / 10%	54 / 20%	**94 / 90%**
put_blocks	48 / 30%	42 / 30%	27 / 0%	**100 / 100%**
classify_items	59 / 20%	**63 / 40%**	36 / 0%	54 / **40%**
press_button	48 / **20%**	13 / 0%	3 / 0%	**58** / 0%
Average SR	22%	20%	4%	**64%**
Total score	248	221	147	**399**

Table 1. Official preliminary ManipArena results. Each cell reports score / success rate. Bold marks the best score or success rate in each row; ties are included.

AgentVLA substantially improves the three execution-style tasks: ring insertion, spoon placement, and block sorting. These tasks benefit from explicit target relations and atomic skill order, while leaving continuous control to the VLA executor.

The two semantic tasks reveal limitations of pure imitation. In classify_items_as_shape, some demonstrations appear to terminate while visible objects remain outside target containers, so the policy can treat a partial state as complete. In press_button_in_order, the semantic objective is a color-order permutation, but training demonstrations mostly contain triangular button layouts whereas some evaluation cases are nearly collinear; this layout shift can induce brittle geometric priors. These failures show that monolithic imitation can entangle task-state estimation, scene grounding, completion detection, and action generation. Once entangled, more demonstrations alone do not provide an explicit check for whether all objects have been handled, whether the layout violates a learned spatial prior, or whether execution should recover before continuing. They also suggest a path beyond simply scaling imitation data. A promising direction is to optimize the agentic layer with task-level feedback, for example through reinforcement learning or preference optimization. Instead of only updating the low-level action generator, the system can learn when to continue a subgoal, switch stages, re-observe the scene, trigger recovery, or re-ground targets under layout shifts. In this view, the VLA executor remains responsible for continuous control, while reinforcement learning improves the planner, verifier, and recovery modules that decide how execution should adapt to task progress and scene changes.

4. Conclusion

These results support a modular view of VLA adaptation. Generalist VLAs are useful executors, but robust manipulation also requires grounding, skill organization, progress monitoring, and recovery. The role of agentization is not to replace the VLA policy, but to atomize long-horizon tasks into separable decisions that should not be hidden inside a single action generator. A planner exposes task structure; grounding binds objects and spatial relations to the scene; a skill layer composes reusable primitives; a verifier decides whether a subgoal is complete; and recovery intervenes when execution stalls. This decomposition makes failures observable and correctable while preserving the learned visuomotor prior of the VLA executor. It also makes the agentic layer a natural target for task-level optimization, while keeping the executor reusable across tasks and embodiments.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, et al. π_0: A vision-language-action flow model for general robot control. *arXiv:2410.24164*, 2024. 1

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv:2212.06817*, 2022.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv:2307.15818*, 2023. 1

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. 2

[5] Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangpin Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, Minghao Zhu, Xiao Lin, Yang Bai, Qian Jiang, Yaxi Zhao, Minghua Zeng, Junlong Gao, Yuming Jiang, Jun Cen, Siteng Huang, et al. RynnBrain: Open embodied foundation models. *arXiv:2602.14979*, 2026. 1

[6] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv:2406.09246*, 2024. 1

[7] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, et al. Open X-embodiment: Robotic learning datasets and RT-X models. *arXiv:2310.08864*, 2023.

[8] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv:2504.16054*, 2025. 1

[9] Yu Sun, Meng Cao, Ping Yang, Rongtao Xu, Yunxiao Yan, Runze Xu, Liang Ma, Roy Gan, Andy Zhai, Qingxuan Chen, Zunnan Xu, Hao Wang, Jincheng Yu, Lucy Liang, Qian Wang, Ivan Laptev, Ian D. Reid, and Xiaodan Liang. ManipArena: Comprehensive real-world evaluation of reasoning-oriented generalist robot manipulation. *arXiv:2603.28545*, 2026. 2

[10] Zhejian Yang, Yongchao Chen, Xueyang Zhou, Jiangyue Yan, Dingjie Song, Yinuo Liu, Yuting Li, Yu Zhang, Pan Zhou, Hechang Chen, and Lichao Sun. Agentic robot: A brain-inspired framework for vision-language-action models in embodied agents. *arXiv:2505.23450*, 2025. 1

[11] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. *arXiv:2602.15922*, 2026. 1

[12] Han Zhao, Jiaxuan Zhang, Wenxuan Song, Pengxiang Ding, and Donglin Wang. VLA^2: Empowering vision-language-action models with an agentic framework for unseen concept manipulation. *arXiv:2510.14902*, 2025. 1