MA-VLA: Multi-Arm Vision-Language-Action Model for Collaboration and Compositional Generalization

Zaibin Zhang[1*], Junlan Xiao[1*], Zhongbo Zhang[1*], Yifan Wang[1], Li Kang[4], Yiran Qin[6], Changxing Xia[1], Heng Zhou[5], Talas Fu[1], Enshen Zhou[7], Ruimao Zhang[3], Zhenfei Yin[2], Huchuan Lu[1], Lijun Wang[1✉]

[1] Dalian University of Technology, [2] University of Oxford, [3] Sun Yat-sen University
[4] Shanghai Jiao Tong University, [5] University of Science and Technology of China
[6] The Chinese University of Hong Kong, Shenzhen, [7] Beihang University

dlutzzb@gmail.com, ljwang@dlut.edu.cn

[*] Equal contribution [✉] Corresponding author

Abstract. Multi-arm collaboration is becoming a core capability in embodied manipulation. Recent vision-language-action (VLA) models integrate perception, language, and control, but most represent language as a single global instruction and do not provide an explicit mechanism for assigning and composing arm-specific behaviors. This design limits transfer to collaboration patterns that differ from those observed during training. We present MA-VLA, a unified framework for multi-arm collaboration via atomic action assignment. MA-VLA decomposes cooperative behavior into mid-level atomic prompts and allocates them to individual arms, enabling explicit subgoal specification and compositional reuse across tasks. To reduce reliance on fixed execution roles, we introduce Arm Shuffle, a training-time permutation of the observation, state, and assigned atomic prompts for each arm. This permutation enforces role-agnostic instruction following and supports recomposition into unseen coordination patterns, which we term multi-arm compositional generalization. We also construct a benchmark in which test-time collaboration patterns are absent in training set. Across simulation and real-world evaluations, prior state-of-the-art VLAs largely fail under these unseen collaborations, while MA-VLA consistently succeeds. These results indicate that structured, per-arm atomic action assignment offers a practical route to scalable generalization in multi-arm embodied systems. Code, models, and data are available at https://github.com/zhangzaibin/future-robots

Keywords: Multi-arm collaboration · Compositional generalization · Vision-language-action

1 Introduction

Embodied Artificial Intelligence (Embodied AI) studies how intelligent systems perceive, reason, and act through continuous interaction with the physical world [13].

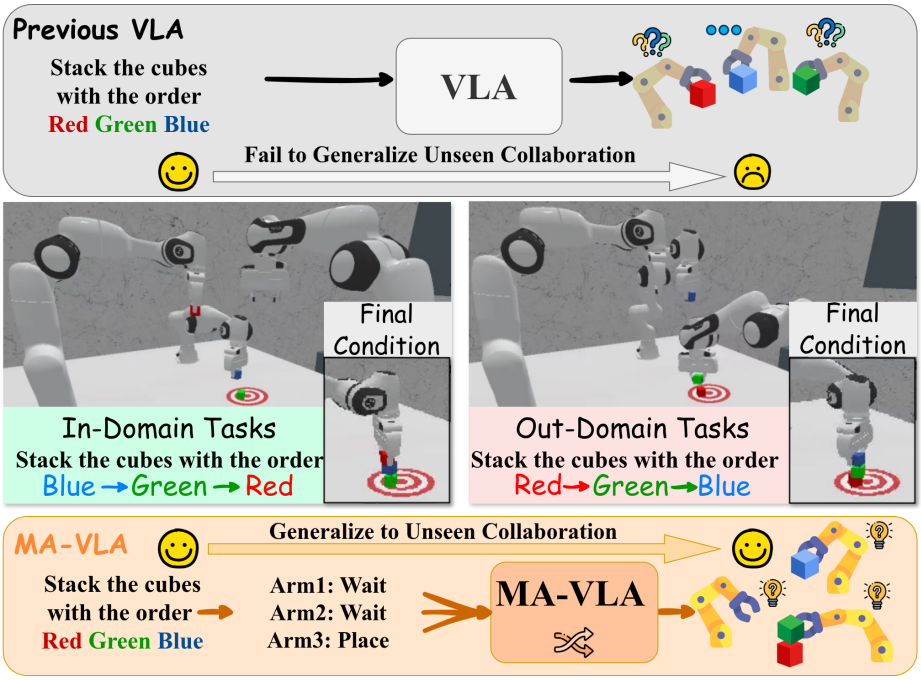

Fig. 1: Comparison with prior VLA systems. Previous VLAs typically execute a task from a single high-level instruction. MA-VLA instead decomposes the instruction into a sequence of interpretable mid-level atomic actions and assigns them to individual arms, which enables recomposition to handle unseen collaboration patterns.

In robotics, this interaction becomes substantially more complex when the setting shifts from single-arm manipulation to multi-arm collaboration [3, 7, 20, 24, 30, 40, 46, 54, 61]. By leveraging parallel execution and coordinated control, multi-arm and multi-robot systems can solve tasks that are infeasible for a single arm, which makes collaboration a central direction in embodied manipulation.

Recent progress has trained dual-arm and multi-arm systems with imitation learning [24, 31, 40, 46, 54] or reinforcement learning [30]. In parallel, end-to-end vision-language-action (VLA) models [41] have enabled dual-arm robots to follow natural-language instructions for manipulation [3, 7, 9, 21, 51, 61]. Despite these advances, most existing VLAs still treat language as a single static command that is mapped to control, without an explicit mechanism for task decomposition and sub-task allocation across arms. This design choice makes collaboration brittle, because division of labor must be inferred implicitly from data and often becomes over-specialized to a small set of training collaboration patterns.

In human teamwork, effective cooperation is driven by division of labor [44]: complex goals are achieved by assigning mid-level responsibilities and executing atomic actions that compose into a coherent plan. This structure is also a natural source of generalization, because new strategies can be formed by recombining

familiar atomic actions with familiar assignment patterns. Motivated by this observation, we study a setting that is largely missing in current multi-arm VLA research: whether a learned system can handle unseen collaboration patterns by recomposing known atomic actions. We refer to this capability as **multi-arm compositional generalization**. However, compositional generalization is non-trivial: test-time coordination often requires unseen combinations of layouts, arm states, and inter-arm dependencies, which cannot be addressed by role swapping or independently trained arms.

To address this gap, we propose MA-VLA, a unified vision-language-action framework for multi-arm collaboration via atomic action assignment. Given a high-level instruction, MA-VLA outputs a sequence of interpretable mid-level atomic actions and assigns them to each arm within a shared model. This representation makes division of labor explicit and offers a simple interface for coordinating execution across arms, while remaining end-to-end at inference time.

To reduce reliance on fixed arm identities and improve compositional transfer, we introduce Arm Shuffle, a training-time permutation that randomizes the mapping between arms and their input bundles, including arm state, local observation, and assigned atomic prompts. By preventing the model from binding behaviors to arm indices, Arm Shuffle promotes role-agnostic instruction following and enables recomposition of atomic actions into unseen collaboration patterns, such as novel stacking orders and object-passing strategies.

We evaluate MA-VLA on RoboFactory [54], RoboTwin 2.0 [9], and a real-world dual-arm SO101 platform under both in-domain settings and multi-arm compositional out-of-domain splits. MA-VLA consistently outperforms competitive imitation and VLA baselines, with the largest gains observed in both in-domain performance and compositional generalization, suggesting a practical path toward flexible and scalable multi-arm collaboration.

2 Related Work

2.1 Multi-arm Robot Manipulation

Behavior Cloning (BC) [11,23,25,28,42,48,49] and Offline Reinforcement Learning (ORL) [8, 26, 29] remain dominant paradigms for robotic policy learning but rely heavily on large-scale expert demonstrations. Reinforcement learning based imitation [17,50] mitigates this dependency yet demands costly online exploration. Recently, generative methods such as Action Chunking Transformer (ACT) [5, 64, 77] and Diffusion Policy [10, 24, 73], empowered by scalable simulation frameworks [22, 46, 47, 55, 59, 81], have shown strong modeling and generalization ability. Most studies [3, 7, 24, 46, 57, 61] still focus on bimanual tasks, while RoboBallet [30] achieves coordination among three or more arms via reinforcement learning, RoboFactory [54] leverages Diffusion Policy for multi-arm collaboration, while research on multi-arm via VLA remains largely unexplored.

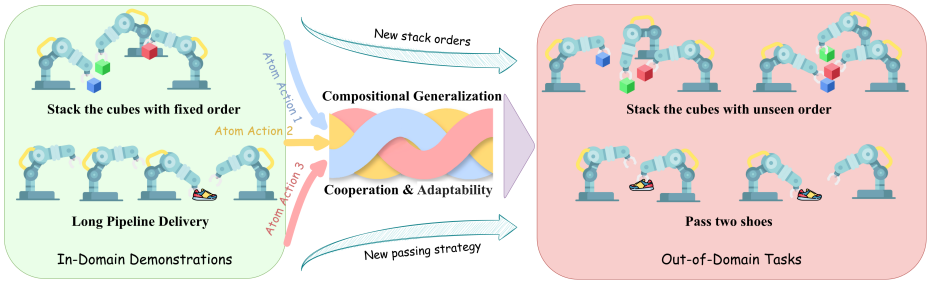

Fig. 2: Multi-arm compositional generalization: unseen collaboration patterns induced by new combinations of arm roles, spatial states, and atomic actions, beyond the patterns observed in training.

2.2 Vision-Language-Action Models

Vision-Language-Action (VLA) models [3,12,19,28,33,35,39,53,56,61,62,65,67, 68, 74–76, 80, 82] bridge perception and control by integrating vision, language, and action generation. With advances in LLMs [1,4,63] and VLMs [27,32,38,58, 60,71,78,79], they leverage cross-modal reasoning to map visual inputs and natural instructions into robot actions. Early systems like RT2 [82] and Octo [62] learn from large-scale multimodal data, while OpenVLA [28] extends pretraining to the Open X-Embodiment dataset [49]. pi0 [3] further introduces flow-matching for continuous control. Yet, these works treat language as static instruction rather than an interactive medium, limiting generalization. MAVLA employs language as an active coordination channel, enabling compositional multi-arm behaviors.

2.3 Task Generalization

Prior studies [9,16,18,35,52,70] primarily address generalization to environmental or visual changes, such as object variations [2, 34, 45]. Recent work extends to task-level generalization through compositional [37, 66, 69, 72] and parameterized policies [14, 15, 43]. Nonetheless, these methods largely focus on single- or dual-arm setups. Our *Arm Shuffle* tackles multi-arm collaborative generalization by randomizing arms' roles during training, breaking fixed dependencies and promoting flexible coordination.

3 Preliminaries

3.1 Vision-Language-Action Models for Multi-arm Manipulation

Single-arm VLA. At timestep t, the robot observes $\mathbf{o}_t = \{\mathbf{I}_t, \mathbf{s}_t\}$, where $\mathbf{I}_t$ is the visual input (e.g., multi-view RGB-D) and $\mathbf{s}_t$ is proprioception (e.g., joint angles, gripper states). Given an instruction $\mathbf{l}$, a policy π_θ predicts the action $\mathbf{a}_t = \pi_\theta(\mathbf{o}_t, \mathbf{l})$. The model is trained via behavioral cloning (BC):

$$\mathcal{L}_{\text{BC}}(\theta) = \mathbb{E}_{(\mathbf{o}_t, \mathbf{l}, \mathbf{a}_t) \sim \mathcal{D}} \big[\ell(\pi_\theta(\mathbf{o}_t, \mathbf{l}), \mathbf{a}_t) \big], \tag{1}$$

where $\ell(\cdot, \cdot)$ is an action-level loss (e.g., MSE or cross-entropy).

Multi-arm extension. For N arms, arm i has local state $\mathbf{s}_t^i$, local view $\mathbf{v}_t^i$, and action $\mathbf{a}_t^i$. A joint policy predicts $\mathbf{A}_t = \{\mathbf{a}_t^1, \ldots, \mathbf{a}_t^N\}$:

$$\mathbf{A}_t = \mathit{\Pi}_\theta(\mathbf{I}_t, \mathbf{s}_t^1, \ldots, \mathbf{s}_t^N, \mathbf{l}). \tag{2}$$

We optimize the summed per-arm BC loss:

$$\mathcal{L}_{\mathrm{BC}}(\theta) = \mathbb{E}_{(\mathbf{O}_t, \mathbf{l}, \mathbf{A}_t) \sim \mathcal{D}} \Big[\sum_{i=1}^{N} \ell\big(\pi_\theta^i(\mathbf{O}_t, \mathbf{l}), \mathbf{a}_t^i\big) \Big], \tag{3}$$

where $\mathbf{O}_t$ aggregates per-arm observations and π_θ^i extracts arm-i actions from $\mathit{\Pi}_\theta$.

3.2 Multi-arm Compositional Generalization

We study multi-arm compositional generalization: solving unseen collaborative executions by recombining the same atomic actions observed in training, without retraining as shown in Figure 2.

Atomic actions. Let $\mathcal{U} = \{u_1, \ldots, u_K\}$ denote a library of atomic actions (e.g., grasp, lift, align, place), shared across arms.

Composing a multi-arm execution. A multi-arm run can be written as a sequence of arm-tagged atomic actions

$$\mathbf{P} = \big[(i_1, u_{k_1}), \ldots, (i_T, u_{k_T})\big], \tag{4}$$

where (i_t, u_{k_t}) means arm i_t executes atomic action u_{k_t} at step t (possibly in parallel with other arms).

Generalization setting. Training covers compositions $\mathcal{P}_{\mathrm{train}}$; at test time we require new compositions $\mathcal{P}_{\mathrm{test}}$:

$$\mathcal{U}_{\mathrm{test}} = \mathcal{U}_{\mathrm{train}}, \qquad \mathcal{P}_{\mathrm{test}} \not\subset \mathcal{P}_{\mathrm{train}}. \tag{5}$$

Thus, the atomic actions remain in-distribution, while the collaboration structure is out-of-distribution. Concretely, the shift comes from new (i) arm-role assignments (i_t), (ii) ordering and synchronization, and (iii) the resulting intermediate states and interactions.

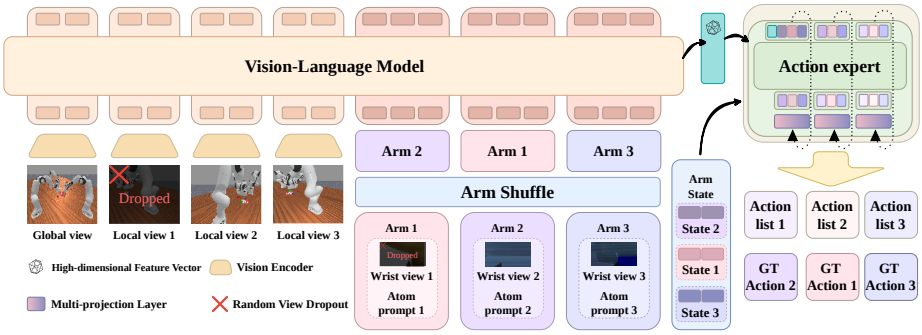

Fig. 3: MA-VLA Pipeline. A high-level instruction is first decomposed into a sequence of atomic prompts. Each arm is represented by a tuple containing its atomic prompt, state, wrist-view observation, and ground-truth action (e.g., **Arm1:** , **Arm2:** , **Arm3:**). During training, *Arm Shuffle* randomly permutes these arm tuples, preventing the model from binding behaviors to fixed arm identities and promoting role-agnostic coordination across diverse collaboration patterns.

4 Method

4.1 System Overview

MA-VLA is a unified architecture for language-guided multi-arm coordination. It consists of two key components: (1) a **VLM-based Planner** that decomposes complex tasks into temporally ordered atomic sub-goals for each arm, and (2) a **VLA Executor** that grounds these linguistic arm-wise sub-goals into actions.

As shown in Figure 5, during execution, the planner interprets a high-level instruction $\mathbf{l}$ and visual observations $\mathbf{I}_t$, producing atomic prompts $\{\mathbf{p}_1, \ldots, \mathbf{p}_T\}$ that specify each arm's behavior at each stage. The VLA executor then conditions on the current scene and state to generate continuous control actions for each arm. This hierarchical language interface lets MA-VLA jointly reason about collaboration structure (planner) and fine-grained physical control (executor). A flow-matching expert [36] further ensures temporally smooth, physically consistent execution across all arms.

4.2 VLM-based Planner: Task Decomposition via Atomic Prompts

To mirror how humans coordinate through language, we introduce a VLM-based planner $\mathcal{P}_\phi$ that converts a high-level goal into a structured sequence of per-arm sub-instructions. Instead of mapping $\mathbf{l}$ directly to low-level control, the planner predicts a stage-wise assignment of atomic prompts by jointly conditioning on the instruction $\mathbf{l}$, visual observations $\mathcal{I}$, and a predefined atomic prompt set $\mathcal{A}$. Here $\mathcal{I}$ denotes a set of images available at planning time, which can come from multi-view observations or consecutive frames (e.g., from a short video

clip), without assuming a specific frame-sampling procedure. This yields an interpretable interface for multi-arm coordination while keeping the downstream executor fully end-to-end.

Formally, the instruction $\mathbf{l}$ is decomposed into T atomic stages:

$$\mathbf{l} = (\mathbf{p}_1, \ldots, \mathbf{p}_T), \qquad \mathbf{p}_t = (p_t^1, \ldots, p_t^N), \tag{6}$$

where $p_t^i \in \mathcal{A}$ denotes the atomic prompt assigned to arm i at stage t (e.g., `grasp object`, `hold bowl`, `place cube`). We use stages to represent temporally grounded sub-goals, each executed until a task-dependent termination condition is met (Appendix).

Given $\mathcal{I}$ and $\mathbf{l}$, the planner generates the full prompt sequence in a single forward pass:

$$(\mathbf{p}1, \ldots, \mathbf{p}T) = \mathcal{P}\phi(\mathcal{I}, \mathbf{l}, \mathcal{A}). \tag{7}$$

Here, $\mathcal{A}$ is a curated and finite set of fine-grained prompt templates shared across tasks, which defines a reproducible action vocabulary (the full list is provided in the supplementary material). In practice, we prompt the VLM to select templates from $\mathcal{A}$ and canonicalize its outputs to valid templates in $\mathcal{A}$, ensuring that each p_t^i corresponds to a well-defined atomic instruction. By reasoning over spatial and semantic cues in $\mathcal{I}$, $\mathcal{P}\phi$ produces temporally coherent and semantically grounded sub-goals that explicitly specify who performs which action at each stage. The planner outputs only language prompts and is not fine-tuned; all control actions are produced by the learned executor. Implementation details, the prompting format, and hyperparameters are provided in the supplementary material.

4.3 VLA Executor: Multi-arm Action Generation

Given the set of atomic prompts produced by the planner, the VLA executor π_θ grounds these linguistic sub-goals into executable control actions through multimodal reasoning. Rather than predicting each arm's motion independently, MA-VLA adopts a unified model that jointly infers all arms' actions within a single forward pass, enabling both coordinated and individualized behaviors.

At each timestep t, the atomic prompts for all arms are concatenated into a unified instruction string:

$$\mathbf{u}_t = \text{Arm0: } p_t^0, \text{ Arm1: } p_t^1, \text{ ... , ArmN: } p_t^N, \tag{8}$$

where $\mathbf{u}_t$ represents the aggregated linguistic context describing the collaborative intent of all arms. This unified prompt serves as the language input to the VLA executor.

The executor then takes as input: (1) multi-view visual observations $\mathbf{I}_t = \mathbf{I}_t^{view}, \mathbf{I}_t^{wrist}$ that include both global and egocentric perspectives, (2) the concatenated proprioceptive states of all arms $\mathbf{s}_t = [\mathbf{s}_t^1, \ldots, \mathbf{s}_t^N]$, and (3) the unified instruction $\mathbf{u}_t$. The VLA executor outputs the complete multi-arm action vector:

$$[\mathbf{a}_t^1, \ldots, \mathbf{a}_t^N] = \pi\theta(\mathbf{I}_t, \mathbf{s}_t, \mathbf{u}_t). \tag{9}$$

By conditioning on the shared visual-linguistic context $\mathbf{u}_t$, MA-VLA effectively captures inter-arm dependencies and ensures coherent coordination among arms, while preserving the flexibility for each arm to perform distinct yet complementary actions.

To produce smooth and physically consistent control trajectories, we adopt a lightweight flow-matching formulation inspired by pi0 [3]. During training, the VLA predicts a denoising vector field that maps noisy latent actions $\mathbf{x}_t^\tau$ toward the ground-truth actions $\mathbf{a}_t$, conditioned on visual observations $\mathbf{I}_t$, proprioceptive states $\mathbf{s}_t$, and the unified linguistic context $\mathbf{u}_t$. At inference time, actions are obtained by integrating the learned flow field from noise to clean latent space. To better support multi-arm collaboration, we further include a multi-head projection layer that separates the shared latent representation into arm-specific action heads.

4.4 Training Strategies for Compositional Generalization

Although MA-VLA learns to execute atomic actions via multimodal conditioning, we find that generalization remains limited when arms are tied to fixed spatial roles or when perception depends heavily on arm-specific states (e.g., visual inputs and proprioceptive states). To address these limitations, we introduce two stochastic regularization strategies that are applied only during training: **Arm Shuffle** and **View Dropout**. Both operate as data-level perturbations that promote role-invariant reasoning and redundancy-aware perception, while leaving the training objective unchanged.

Arm Shuffle. In each training iteration, with a probability p_{shuffle}, we randomly permute the correspondence between arms' states, views, prompts, and actions:

$$(\mathbf{s}_t^i, \mathbf{v}_t^i, \mathbf{p}_t^i, \mathbf{a}_t^i) \xrightarrow{\text{shuffle}} (\mathbf{s}_t^{\sigma(i)}, \mathbf{v}_t^{\sigma(i)}, \mathbf{p}_t^{\sigma(i)}, \mathbf{a}_t^{\sigma(i)}), \tag{10}$$

where $\sigma \in \mathcal{S}_N$ is a random permutation of the N arm indices. This stochastic role reassignment prevents the model from overfitting to fixed positional or identity-based correlations, forcing each arm to interpret its atomic prompt semantically rather than structurally.

View Dropout. To enhance perception robustness, we apply random masking to a subset of visual inputs with probability p_{drop}. Let $\mathcal{M} \subseteq \{1, \ldots, M\}$ denote the indices of dropped camera views. We define the masked observation as

$$\tilde{\mathbf{I}}_t = \text{Mask}(\mathbf{I}_t, \mathcal{M}), \tag{11}$$

where $\text{Mask}(\cdot)$ replaces selected views with zeros. This encourages the model to utilize redundant spatial cues and learn consistent multi-view reasoning.

Training objective. Both strategies are integrated as stochastic augmentations under a unified behavioral cloning objective. At each iteration, the model samples whether to apply shuffle or dropout, and computes the same action-level loss:

$$\mathcal{L}_{\mathrm{t}}(\theta) = \mathbb{E}_{(\mathbf{O}_t, \mathbf{l}, \mathbf{A}_t)\sim\mathcal{D}} \left[\sum_{i=1}^{N} \ell\Big(\pi_\theta(\tilde{\mathbf{I}}_t, \mathbf{s}_t^i, \mathbf{p}_t^i), \mathbf{a}_t^i \Big) \right]. \tag{12}$$

where $\tilde{\mathbf{I}}_t$ and $\mathbf{s}_t^i, \mathbf{p}t^i, \mathbf{a}t^i$ denote the possibly perturbed inputs after applying stochastic *Arm Shuffle* and *View Dropout*, controlled respectively by the shuffle rate p_{shuffle} and the view-drop rate p_{drop}. These perturbations preserve the original learning objective while implicitly regularizing MA-VLA toward permutation-invariant coordination and more robust visual grounding.

5 Experiments

In this section, we evaluate MA-VLA under both in-domain settings and multi-arm compositional generalization (out-of-domain) splits.

5.1 Experimental Setup

Training configurations. We report two training configurations. **Clean** disables Arm Shuffle and View Dropout, training on the original demonstrations. **Regularized** enables Arm Shuffle and View Dropout during training.

Benchmarks. We evaluate two simulation benchmarks and one real-robot platform, reporting in-domain performance and out-of-domain compositional generalization. **RoboFactory** [54] contains cooperative manipulation tasks with 2–4 arms, emphasizing coordination and parallel execution. **RoboTwin2.0** [9] focuses on dual-arm collaboration under stronger visual disturbances, including additional distractors, background variation, and lighting changes; we use standard multi-view observations with wrist-camera views. **Real-world SO101** is a dual-arm system with 12 degrees of freedom (6 per arm), equipped with one global RGB camera and one wrist camera per arm; all cameras run at 30Hz. Real-robot tasks include Stack Bowls, Place Cubes, Pass Toys, and Stack Cubes; details are in the supplementary material.

Demonstrations and atomic action labels. For each simulation task, we collect 150 expert demonstrations in the in-domain setting. Each trajectory contains synchronized multi-view observations, per-arm proprioception, and action sequences. We convert demonstrations into frame-level atomic action prompts using a rule-based parser with task-specific state predicates (e.g., contact, grasp status, object pose thresholds), producing training tuples $(\mathbf{O}_t, \mathbf{l}_t, \mathbf{A}_t)$ where $\mathbf{l}_t$ is the atomic prompt at time t.

Evaluation protocol and metrics. We evaluate two settings. **In-domain Collaboration** tests the same task distribution as training with randomized perturbations (object pose shifts, camera viewpoint changes, and environment variations). **Out-of-domain Compositional Generalization** tests unseen collaboration structures where atomic actions remain in-distribution but are recomposed into unseen temporal orders, role assignments, or interaction patterns. For simulation, each configuration is evaluated over 100 rollouts and we report mean success rate; success is defined by task-specific goal predicates in the simulator. Unless otherwise stated, we use the same perturbation suite across methods. For compositional generalization splits, we enable Arm Shuffle and View Dropout during training; otherwise we train on the original demonstrations.

Real-world data and evaluation. On SO101, we evaluate four bimanual tasks: Stack Two Bowls, Place Two Cubes, Pass Two Toys, and Stack Two Cubes. The detailed real-robot task settings are provided in the supplementary material. For each task, we collect 50 teleoperated demonstrations using the LeRobot [6], train for 15,000 gradient steps with a batch size of 32, and evaluate over 20 episodes. We report the success rate using the same in-domain and out-of-domain split definition as in simulation. Each trial starts from a randomized but valid initial configuration and stops upon success, timeout, or a safety stop.

Backbones and baselines. Our framework uses Pi0 [3] initialized from the official `pi0_base` checkpoint. The VLM planner uses GPT-4.1 to generate atomic action prompts (see supplementary material for the prompt template and decoding settings). We compare against representative baselines: ACT [77], DP [10], DP3 [73], and Pi0-FAST [51]. Since prior VLA policies are designed for single-arm inputs, we adapt their input/output interfaces to multi-arm RoboFactory and retrain them from the same demonstrations with matched training budgets.

5.2 Implementation Details

All experiments use two NVIDIA A800 GPUs with batch size 32. For RoboFactory, we train 2-arm tasks for 10,000 steps and 3–4 arm tasks for 15,000 steps. For RoboTwin 2.0 (Hard), we train for 30,000 steps. The horizon is 50 for all tasks. For the compositional setting, we set Arm Shuffle probability and View Dropout probability per out-of-domain task: Three Robots Stack Cube (Reordering): p_{shuffle}=0.8, p_{drop}=0.2; Pass Two Shoes (Extended Collaboration): p_{shuffle}=1.0, p_{drop}=0.4; Reverse Stack Bowls (Reordering+Disturbance): p_{shuffle}=0.6, p_{drop}=0.4. For SO101, we enable Arm Shuffle throughout training to reduce arm-identity bias and encourage instruction-centric coordination. Specifically, we set p_{shuffle}=1.0 and apply View Dropout (masking) with probability p_{drop}=0.4 on multi-camera observations. All other training hyperparameters follow the simulation setup unless otherwise stated. We keep all other hyperparameters identical across methods within each benchmark.

Table 1: Performance on RoboFactory Benchmark [54]. Abbrev.: Lift=Lift Barrier, Place=Place Food, Stack2=Two Arms Stack Cube, Pass=Pass Shoe; Stack3=Three Arms Stack Cube, Align=Camera Alignment, Deliver=Long Pipeline Delivery, Photo=Take Photo.

Two-arm tasks					
Method	Lift	Place	Stack2	Pass	Avg
DP [10]	58.0%	20.0%	20.0%	12.0%	27.5%
Pi0-Fast [51]	90.0%	27.0%	50.0%	63.0%	57.5%
Pi0 [3]	97.0%	58.0%	82.0%	84.0%	80.3%
MA-VLA	**98.0%**	**59.0%**	**86.0%**	**91.0%**	**83.5%**
	Three-arm Tasks		Four-arm Tasks		
Method	Stack3	Align	Deliver	Photo	Avg
DP [10]	22.0%	19.0%	0.0%	20.0%	15.3%
Pi0-Fast [51]	3.0%	22.0%	10.0%	12.0%	11.8%
Pi0 [3]	48.0%	94.0%	82.0%	**82.0%**	76.5%
MA-VLA	**58.0%**	**96.0%**	**97.0%**	**82.0%**	**83.3%**

Table 2: Performance on RoboTwin 2.0 (Hard) [9]. Abbrev.: Shoes=Place Dual Shoes; Handover=Handover Block; Bowls=Stack Bowls Two; Roller=Grab Roller; Cabinet=Place Object (Cabinet); Skillet=Place Bread (Skillet); Stamp=Stamp Seal.

Method	Shoes	Handover	Bowls	Roller	Cabinet	Skillet	Stamp	Avg
ACT [77]	0.0	0.0	0.0	58.0	23.0	5.0	0.0	12.3
DP [10]	4.0	21.0	1.0	58.0	42.0	6.0	0.0	18.9
DP3 [73]	7.0	14.0	39.0	78.0	52.0	9.0	1.0	28.6
Pi0-FAST [51]	10.0	25.0	44.0	83.0	23.0	2.0	2.0	27.0
Pi0 [3]	14.0	27.0	59.0	98.0	59.0	28.0	3.0	41.1
MA-VLA	**22.0**	**46.0**	**63.0**	**99.0**	**71.0**	**30.0**	**12.0**	**49.0**

5.3 In-domain Collaboration Results

Table 1 shows that conditioning on atomic action instructions consistently improves in-domain collaboration on RoboFactory. The gains grow with more arms (3–4 arms), suggesting atomic prompts better disambiguate per-arm responsibilities as coordination complexity increases. On RoboTwin2.0 (Hard), Table 2 shows MA-VLA remains robust under strong visual disturbances, consistently improving over the Pi0 backbone and diffusion-based baselines.

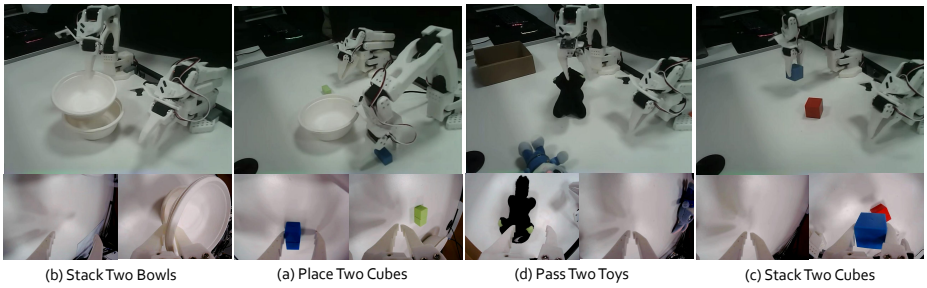

Fig. 4: Setup of the real-world dual-arm SO101 tasks. Top: global view. Bottom: wrist views (left: left-wrist camera; right: right-wrist camera).

5.4 Out-of-domain Compositional Generalization Results

We evaluate compositional generalization across unseen collaboration patterns in RoboFactory and RoboTwin2.0 (Hard). As shown in Table 3, MA-VLA achieves non-trivial success where end-to-end imitation baselines collapse, improving out-of-domain success by up to 13.0These scenarios contain novel role orders and coordination structures unseen in training. Overall, atomic action conditioning, Arm Shuffle, and View Dropout reduce reliance on fixed arm identities and promote instruction-centric coordination, enabling recomposition of learned atomic skills under unseen collaboration paradigms.

Table 3: Results on unseen out-of-domain collaboration tasks. Arrows denote the required stacking order of cube colors. Notation: GBR=G→B→R; RGB=R→G→B; BRG=B→R→G.

Method	GBR	RGB	BRG	Pass Two Shoes	Stack Two Bowls	Avg
DP [10]	0.0	0.0	0.0	0.0	0.0	0.0
Pi0-FAST [51]	0.0	0.0	0.0	0.0	0.0	0.0
Pi0 [3]	0.0	0.0	0.0	0.0	0.0	0.0
MA-VLA	**28.0**	**9.0**	**9.0**	**9.0**	**10.0**	**13.0**

5.5 Real-world Dual SO101 Results

Table 4 summarizes real-world results. MA-VLA consistently improves Pi0 in-domain, and importantly achieves non-zero success in role-reversed out-of-domain settings where Pi0 fails. This suggests that atomic action conditioning, together with Arm Shuffle and View Dropout, mitigates arm-identity bias and improves transfer to unseen real-world collaboration patterns.

Table 4: Real-world results on dual-arm SO101. Each task is evaluated in-domain (ID) and compositional out-of-domain (OOD). OOD reconfigures the program: Stack Bowls/Stack Cubes use reversed stacking, Place Cubes uses reversed placement, and Pass Toys uses an alternative passing order.

	Stack Bowls		Place Cubes		Pass Toys		Stack Cubes	
Method	ID	OOD	ID	OOD	ID	OOD	ID	OOD
Pi0 [3]	9/20	0/20	12/20	0/20	3/20	0/20	5/20	0/20
MA-VLA	**12/20**	**10/20**	**15/20**	**8/20**	**6/20**	**2/20**	**8/20**	**2/20**

6 Ablation Study

We use the Three Robots Stack Cube unseen-order task for evaluation. The in-domain order is blue–green–red, while the out-of-domain configurations include green–blue–red (G! →B! →R), red–green–blue (R! →G! →B), and blue–red–green (B! →R! →G). For out-of-domain results, we average performance across the three unseen configurations.

6.1 Ablation of Components

We ablate our components on the Three Robots Stack Cube unseen-order task. As shown in Table 5, introducing atomic actions markedly improves in-domain performance. Adding Arm Shuffle yields a breakthrough in out-of-domain generalization, achieving non-zero success where prior models fail and demonstrating clearer compositional generalization. Since generalization involves substantial visual discrepancies (e.g., environment changes, target appearance, and arm configurations), we also incorporate View Dropout. This further boosts out-of-domain compositional generalization with only minimal in-domain degradation.

Table 5: Ablation Experiment of Components.

Atom	Shuffle	View Dropout	Out-of-domain	In-domain
			0.0%	48.0%
✓			0.0%	**58.0%**
✓	✓		7.3%	52.0%
✓	✓	✓	**15.3%**	53.0%

6.2 Ablation of Shuffle Probability

We further analyze the impact of the shuffle rate through a controlled ablation study, where p_{shuffle} determines the probability of randomly permuting

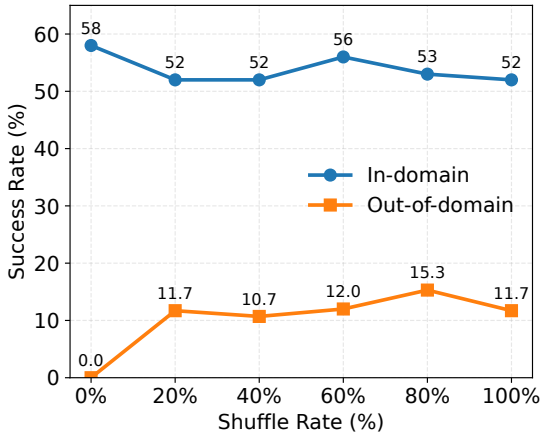

Fig. 5: Effect of shuffle rate on in- and out-of-domain settings.

arms' states, viewpoints, and instructions. As shown in Figure 5, we vary the shuffle probability from 0% to 100%. Increasing the shuffle rate boosts out-of-domain performance—from near-zero to significantly higher levels—highlighting its strong effect on compositional generalization. While in-domain accuracy experiences a mild drop, it remains stable. This indicates that shuffle enables the model to gradually generalize better to new collaboration tasks.

6.3 Ablation of Separate Model

For multi-arm control, a simple baseline trains an independent VLA per arm. To build this setting, we split multi-arm trajectories into per-arm streams so each model sees only its own viewpoint, state, and action history, and we annotate arm-specific atomic actions for single-arm instruction following. At test time, we run three separate VLA models in parallel, one per arm. Results in Table 6 show poor compositional generalization: while each arm follows its atomic actions, the models fail to coordinate on unseen multi-arm tasks. This design also introduces inference and deployment overhead that grows linearly with the number of arms, underscoring the limitations of independent-arm training and the scalability of our unified framework.

Table 6: Comparison between MA-VLA and Separate Model.

Method	Out-of-domain	In-domain	Avg.	VLA Number
Pi0	0.0%	48.0%	24.0%	**1**
Separate Pi0	0.0%	**61.0%**	30.5%	3
MA-VLA	**15.3%**	53.0%	**34.2%**	**1**

7 Visualization

Figure 6 shows MA-VLA rollouts in RoboFactory (simulation) and on the real-world SO101 under both in-domain and out-of-domain splits. The out-of-domain split targets *multi-arm compositional generalization*: the model must recombine seen atomic actions into unseen multi-arm coordination patterns.

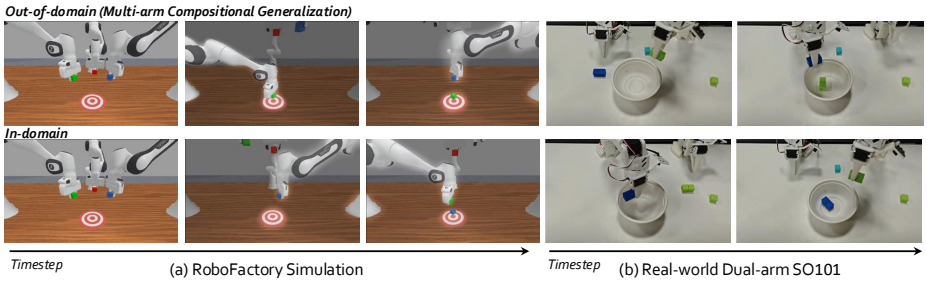

Fig. 6: Qualitative rollouts of MA-VLA in RoboFactory and on SO101 under in-domain and out-of-domain splits. Out-of-domain evaluates compositional generalization by requiring unseen recombinations of atomic actions under novel arm and object states.

8 Conclusion

We introduce MA-VLA, a unified VLA framework that makes multi-arm collaboration explicit by decomposing a high-level instruction into interpretable atomic actions and assigning them to individual arms within a single model. With Arm Shuffle to encourage role-agnostic behavior, MA-VLA improves compositional transfer to unseen collaboration patterns. Experiments on RoboFactory, RoboTwin 2.0, and real-world SO101 show consistent gains over strong imitation learning and VLA baselines in both in-domain performance and multi-arm compositional generalization, supporting a practical route to scalable multi-arm instruction following.

9 Acknowledgment

This paper is supported by the National Natural Science Foundation of China (U23A20386, 62422610, 62441231, 62276045, 62576072), Liao Ning Science and Technology Plan (2025JH2/101330121, 2025JH2/101330124, 2023JH26/10200016), and Dalian City Science and Technology Innovation Fund (2023JJ11CG001).

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Akinola, I., Xu, J., Carius, J., Fox, D., Narang, Y.: Tacsl: A library for visuotactile sensor simulation and learning. IEEE Transactions on Robotics (2025)
3. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: pi_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
5. Buamanee, T., Kobayashi, M., Uranishi, Y., Takemura, H.: Bi-act: Bilateral control-based imitation learning via action chunking with transformer. In: 2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM). pp. 410–415. IEEE (2024)
6. Cadene, R., Aliberts, S., Capuano, F., Aractingi, M., Zouitine, A., Kooijmans, P., Choghari, J., Russi, M., Pascal, C., Palma, S., et al.: Lerobot: An open-source library for end-to-end robot learning. arXiv preprint arXiv:2602.22818 (2026)
7. Cheang, C., Chen, S., Cui, Z., Hu, Y., Huang, L., Kong, T., Li, H., Li, Y., Liu, Y., Ma, X., et al.: Gr-3 technical report. arXiv preprint arXiv:2507.15493 (2025)
8. Chebotar, Y., Vuong, Q., Hausman, K., Xia, F., Lu, Y., Irpan, A., Kumar, A., Yu, T., Herzog, A., Pertsch, K., et al.: Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In: Conference on Robot Learning. pp. 3909–3928. PMLR (2023)
9. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088 (2025)
10. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research **44**(10-11), 1684–1704 (2025)
11. Dalal, M., Mandlekar, A., Garrett, C., Handa, A., Salakhutdinov, R., Fox, D.: Imitating task and motion planning with visuomotor transformers. arXiv preprint arXiv:2305.16309 (2023)
12. Driess, D., Springenberg, J.T., Ichter, B., Yu, L., Li-Bell, A., Pertsch, K., Ren, A.Z., Walke, H., Vuong, Q., Shi, L.X., et al.: Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. arXiv preprint arXiv:2505.23705 (2025)
13. Duan, J., Yu, S., Tan, H.L., Zhu, H., Tan, C.: A survey of embodied ai: From simulators to research tasks. IEEE Transactions on Emerging Topics in Computational Intelligence **6**(2), 230–244 (2022)
14. Garcia, R., Chen, S., Schmid, C.: Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 8996–9002. IEEE (2025)
15. Gong, R., Huang, J., Zhao, Y., Geng, H., Gao, X., Wu, Q., Ai, W., Zhou, Z., Terzopoulos, D., Zhu, S.C., et al.: Arnold: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20483–20495 (2023)

16. Gupta, A., Kumar, V., Lynch, C., Levine, S., Hausman, K.: Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. arXiv preprint arXiv:1910.11956 (2019)
17. Ho, J., Ermon, S.: Generative adversarial imitation learning. Advances in neural information processing systems **29** (2016)
18. Hua, P., Liu, M., Macaluso, A., Lin, Y., Zhang, W., Xu, H., Wang, L.: Gensim2: Scaling robot data generation with multi-modal and reasoning llms. arXiv preprint arXiv:2410.03645 (2024)
19. Huang, H., Liu, F., Fu, L., Wu, T., Mukadam, M., Malik, J., Goldberg, K., Abbeel, P.: Otter: A vision-language-action model with text-aware visual feature extraction. arXiv preprint arXiv:2503.03734 (2025)
20. Im, H., Jeong, E., Fu, J., Kolobov, A., Lee, Y.: Twinvla: Data-efficient bimanual manipulation with twin single-arm vision-language-action models. arXiv preprint arXiv:2511.05275 (2025)
21. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: pi0.5: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
22. James, S., Ma, Z., Arrojo, D.R., Davison, A.J.: Rlbench: The robot learning benchmark & learning environment. IEEE Robotics and Automation Letters **5**(2), 3019–3026 (2020)
23. Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., Finn, C.: Bc-z: Zero-shot task generalization with robotic imitation learning. In: Conference on Robot Learning. pp. 991–1002. PMLR (2022)
24. Jiang, J.J., Wu, X.M., He, Y.X., Zeng, L.A., Wei, Y.L., Zhang, D., Zheng, W.S.: Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework. arXiv preprint arXiv:2503.09186 (2025)
25. Jiang, Y., Gupta, A., Zhang, Z., Wang, G., Dou, Y., Chen, Y., Fei-Fei, L., Anandkumar, A., Zhu, Y., Fan, L.: Vima: General robot manipulation with multimodal prompts. arXiv preprint arXiv:2210.03094 **2**(3), 6 (2022)
26. Kalashnikov, D., Varley, J., Chebotar, Y., Swanson, B., Jonschkowski, R., Finn, C., Levine, S., Hausman, K.: Mt-opt: Continuous multi-task robotic reinforcement learning at scale. arXiv preprint arXiv:2104.08212 (2021)
27. Karamcheti, S., Nair, S., Balakrishna, A., Liang, P., Kollar, T., Sadigh, D.: Prismatic vlms: Investigating the design space of visually-conditioned language models. In: Forty-first International Conference on Machine Learning (2024)
28. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
29. Kumar, A., Singh, A., Ebert, F., Nakamoto, M., Yang, Y., Finn, C., Levine, S.: Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. arXiv preprint arXiv:2210.05178 (2022)
30. Lai, M., Go, K., Li, Z., Kröger, T., Schaal, S., Allen, K., Scholz, J.: Roboballet: Planning for multirobot reaching with graph neural networks and reinforcement learning. Science Robotics **10**(106), eads1204 (2025)
31. Lee, A., Chuang, I., Chen, L.Y., Soltani, I.: Interact: Inter-dependency aware action chunking with hierarchical attention transformers for bimanual manipulation. arXiv preprint arXiv:2409.07914 (2024)
32. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International conference on machine learning. pp. 12888–12900. PMLR (2022)

33. Li, P., Chen, Y., Wu, H., Ma, X., Wu, X., Huang, Y., Wang, L., Kong, T., Tan, T.: Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. arXiv preprint arXiv:2506.07961 (2025)
34. Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., et al.: Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941 (2024)
35. Li, Y., Deng, Y., Zhang, J., Jang, J., Memmel, M., Yu, R., Garrett, C.R., Ramos, F., Fox, D., Li, A., et al.: Hamster: Hierarchical action models for open-world robot manipulation. arXiv preprint arXiv:2502.05485 (2025)
36. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
37. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)
38. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
39. Liu, J., Chen, H., An, P., Liu, Z., Zhang, R., Gu, C., Li, X., Guo, Z., Chen, S., Liu, M., et al.: Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. arXiv preprint arXiv:2503.10631 (2025)
40. Lu, G., Yu, T., Deng, H., Chen, S.S., Tang, Y., Wang, Z.: Anybimanual: Transferring unimanual policy for general bimanual manipulation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13662–13672 (2025)
41. Ma, Y., Song, Z., Zhuang, Y., Hao, J., King, I.: A survey on vision-language-action models for embodied ai. arXiv preprint arXiv:2405.14093 (2024)
42. Mandlekar, A., Xu, D., Martín-Martín, R., Savarese, S., Fei-Fei, L.: Learning to generalize across long-horizon tasks from human demonstrations. arXiv preprint arXiv:2003.06085 (2020)
43. Mees, O., Hermann, L., Rosete-Beas, E., Burgard, W.: Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. IEEE Robotics and Automation Letters **7**(3), 7327–7334 (2022)
44. Minsky, M.: Society of mind. Simon and Schuster (1986)
45. Mu, T., Ling, Z., Xiang, F., Yang, D., Li, X., Tao, S., Huang, Z., Jia, Z., Su, H.: Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. arXiv preprint arXiv:2107.14483 (2021)
46. Mu, Y., Chen, T., Peng, S., Chen, Z., Gao, Z., Zou, Y., Lin, L., Xie, Z., Luo, P.: Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In: European Conference on Computer Vision. pp. 264–273. Springer (2024)
47. Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., Zhu, Y.: Robocasa: Large-scale simulation of everyday tasks for generalist robots. arXiv preprint arXiv:2406.02523 (2024)
48. Osa, T., Pajarinen, J., Neumann, G., Bagnell, J.A., Abbeel, P., Peters, J., et al.: An algorithmic perspective on imitation learning. Foundations and Trends® in Robotics **7**(1-2), 1–179 (2018)
49. O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 6892–6903. IEEE (2024)
50. Papagiannis, G., Li, Y.: Imitation learning with sinkhorn distances. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 116–131. Springer (2022)

51. Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: Fast: Efficient action tokenization for vision-language-action models. arXiv preprint arXiv:2501.09747 (2025)
52. Pumacay, W., Singh, I., Duan, J., Krishna, R., Thomason, J., Fox, D.: The colosseum: A benchmark for evaluating generalization for robotic manipulation. arXiv preprint arXiv:2402.08191 (2024)
53. Qi, Z., Zhang, W., Ding, Y., Dong, R., Yu, X., Li, J., Xu, L., Li, B., He, X., Fan, G., et al.: Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. arXiv preprint arXiv:2502.13143 (2025)
54. Qin, Y., Kang, L., Song, X., Yin, Z., Liu, X., Liu, X., Zhang, R., Bai, L.: Robofactory: Exploring embodied agent collaboration with compositional constraints. arXiv preprint arXiv:2503.16408 (2025)
55. Qin, Y., Shi, Z., Yu, J., Wang, X., Zhou, E., Li, L., Yin, Z., Liu, X., Sheng, L., Shao, J., et al.: Worldsimbench: Towards video generation models as world simulators. arXiv preprint arXiv:2410.18072 (2024)
56. Qu, D., Song, H., Chen, Q., Yao, Y., Ye, X., Ding, Y., Wang, Z., Gu, J., Zhao, B., Wang, D., et al.: Spatialvla: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830 (2025)
57. Shi, L.X., Ichter, B., Equi, M., Ke, L., Pertsch, K., Vuong, Q., Tanner, J., Walling, A., Wang, H., Fusai, N., et al.: Hi robot: Open-ended instruction following with hierarchical vision-language-action models. arXiv preprint arXiv:2502.19417 (2025)
58. Tan, H., Zhou, E., Li, Z., Xu, Y., Ji, Y., Chen, X., Chi, C., Wang, P., Jia, H., Ao, Y., et al.: Robobrain 2.5: Depth in sight, time in mind. arXiv preprint arXiv:2601.14352 (2026)
59. Tao, S., Xiang, F., Shukla, A., Qin, Y., Hinrichsen, X., Yuan, X., Bao, C., Lin, X., Liu, Y., Chan, T.k., et al.: Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. arXiv preprint arXiv:2410.00425 (2024)
60. Team, B.R., Cao, M., Tan, H., Ji, Y., Chen, X., Lin, M., Li, Z., Cao, Z., Wang, P., Zhou, E., et al.: Robobrain 2.0 technical report. arXiv preprint arXiv:2507.02029 (2025)
61. Team, G.R., Abeyruwan, S., Ainslie, J., Alayrac, J.B., Arenas, M.G., Armstrong, T., Balakrishna, A., Baruch, R., Bauza, M., Blokzijl, M., et al.: Gemini robotics: Bringing ai into the physical world. arXiv preprint arXiv:2503.20020 (2025)
62. Team, O.M., Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., et al.: Octo: An open-source generalist robot policy. arXiv preprint arXiv:2405.12213 (2024)
63. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
64. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
65. Wang, Y., Zhu, H., Liu, M., Yang, J., Fang, H.S., He, T.: Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. arXiv preprint arXiv:2507.01016 (2025)
66. Wang, Y.R., Ung, C., Tannert, G., Duan, J., Li, J., Le, A., Oswal, R., Grotz, M., Pumacay, W., Deng, Y., et al.: Roboeval: Where robotic manipulation meets structured and scalable evaluation. arXiv preprint arXiv:2507.00435 (2025)
67. Wen, J., Zhu, Y., Li, J., Tang, Z., Shen, C., Feng, F.: Dexvla: Vision-language model with plug-in diffusion expert for general robot control. arXiv preprint arXiv:2502.05855 (2025)

68. Wu, Z., Zhou, Y., Xu, X., Wang, Z., Yan, H.: Momanipvla: Transferring vision-language-action models for general mobile manipulation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1714–1723 (2025)
69. Yang, R., Chen, H., Zhang, J., Zhao, M., Qian, C., Wang, K., Wang, Q., Koripella, T.V., Movahedi, M., Li, M., et al.: Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. arXiv preprint arXiv:2502.09560 (2025)
70. Yenamandra, S., Ramachandran, A., Yadav, K., Wang, A., Khanna, M., Gervet, T., Yang, T.Y., Jain, V., Clegg, A.W., Turner, J., et al.: Homerobot: Open-vocabulary mobile manipulation. arXiv preprint arXiv:2306.11565 (2023)
71. Yu, S., Chen, Y., Ju, H., Jia, L., Zhang, F., Huang, S., Wu, Y., Cui, R., Ran, B., Zhang, Z., et al.: How far are vlms from visual spatial intelligence? a benchmark-driven perspective. arXiv preprint arXiv:2509.18905 (2025)
72. Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., Levine, S.: Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Conference on robot learning. pp. 1094–1100. PMLR (2020)
73. Ze, Y., Zhang, G., Zhang, K., Hu, C., Wang, M., Xu, H.: 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. arXiv preprint arXiv:2403.03954 (2024)
74. Zhang, J., Chen, Y., Xu, Y., Huang, Z., Zhou, Y., Yuan, Y.J., Cai, X., Huang, G., Quan, X., Xu, H., et al.: 4d-vla: Spatiotemporal vision-language-action pretraining with cross-scene calibration. arXiv preprint arXiv:2506.22242 (2025)
75. Zhang, W., Liu, H., Qi, Z., Wang, Y., Yu, X., Zhang, J., Dong, R., He, J., Lu, F., Wang, H., et al.: Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. arXiv preprint arXiv:2507.04447 (2025)
76. Zhao, Q., Lu, Y., Kim, M.J., Fu, Z., Zhang, Z., Wu, Y., Li, Z., Ma, Q., Han, S., Finn, C., et al.: Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1702–1713 (2025)
77. Zhao, T.Z., Kumar, V., Levine, S., Finn, C.: Learning fine-grained bimanual manipulation with low-cost hardware. arXiv preprint arXiv:2304.13705 (2023)
78. Zhou, E., An, J., Chi, C., Han, Y., Rong, S., Zhang, C., Wang, P., Wang, Z., Huang, T., Sheng, L., et al.: Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. Advances in Neural Information Processing Systems **38**, 28404–28481 (2026)
79. Zhou, E., Chi, C., Li, Y., An, J., Zhang, J., Rong, S., Han, Y., Ji, Y., Liu, M., Wang, P., et al.: Robotracer: Mastering spatial trace with reasoning in vision-language models for robotics. arXiv preprint arXiv:2512.13660 (2025)
80. Zhou, Z., Zhu, Y., Wen, J., Shen, C., Xu, Y.: Vision-language-action model with open-world embodied reasoning from pretrained knowledge. arXiv preprint arXiv:2505.21906 (2025)
81. Zhu, Y., Wong, J., Mandlekar, A., Martín-Martín, R., Joshi, A., Nasiriany, S., Zhu, Y.: robosuite: A modular simulation framework and benchmark for robot learning. arXiv preprint arXiv:2009.12293 (2020)
82. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023)